ARMY RESEARCH LABORATORY

Parametric Study of Beta-Endpoint Energy in Direct Energy Converters

by Kara Blaine and Marc Litz

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Adelphi, MD 20783-1197

ARL-TR-4034   **January 2007**

Parametric Study of Beta-Endpoint Energy in Direct Energy Converters

Kara Blaine and Marc Litz
Sensors and Electron Devices Directorate, ARL

REPORT DOCUMENTATION PAGE		*Form Approved* OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
January 2007	Progress	May 2006

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Parametric Study of Beta-Endpoint Energy in Direct Energy Converters	
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
Kara Blaine and Marc Litz	
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
U.S. Army Research Laboratory ATTN: AMSRD-ARL-SE-DE 2800 Powder Mill Road Adelphi, MD 20783-1197	ARL-TR-4034

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
U.S. Army Research Laboratory 2800 Powder Mill Road Adelphi, MD 20783-1197	
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Several solid-state materials have been identified for potential use as direct-energy-converter (DEC) for isotope-based batteries. Numerical simulations, using a nuclear scattering code (MCNPX), have been performed to determine the electron energy deposited in the material. Two different parametric studies were performed 1) varying β-endpoint energy of a spectrum illuminating layers of silicon-carbide (SiC), 2) the other varying the material layers while keeping the β-endpoint energy constant. The goal of the simulations is to identify the regions within the materials of maximum energy deposition so DEC devices can be fabricated for higher efficiency. The results show that 50 keV and 100 keV β-endpoint energies stand to have most impact to future Schottky devices, generating the largest number density of carriers, and highest energy deposition efficiency in the first 10 μm of SiC.

15. SUBJECT TERMS

SiC, DEC, isotope

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Marc Litz
a. REPORT	b. ABSTRACT	c. THIS PAGE	SAR	60	
Unclassified	Unclassified	Unclassified			19b. TELEPHONE NUMBER (Include area code) 301-394-5556

Contents

List of Figures

List of Tables

INTENTIONALLY LEFT BLANK

1. Background

Direct energy conversion in isotope batteries can yield steady low-power over ten to twenty years without significant degradation. Low-power sensors can benefit from nuclear batteries by harvesting from the high-density storage. The energy density of isotopes is 5 to 9 orders of magnitude more dense than the energy stored in chemical bonds. With isotope batteries, the opportunity exists for sensors to be left unattended for the duration of their lifetimes-decades for some applications. Chemical batteries cannot last that long, even unused, because of degradation of the chemistry. Low maintenance of the unattended sensors keeps them unobtrusive. Unattended underwater sensors, which can be dropped into the ocean, can also benefit from long-lived, low-maintenance isotope batteries. Replacement issues in the field would be non-existent. Embedded sensors would also be feasible, such as sensors buried in the earth which could detect seismic pulses or sensors built into building foundations or bridges which could detect structural damage.

Chemical batteries vary depending upon the chemistry. However their basic components are all the same. They have a positive anode and a negative cathode with an electrolytic separator between them. Electrons and holes flow through the electrolyte. The difference between single use batteries and rechargeable batteries lies in whether the electrochemical reaction that occurs is reversible or not. If the reaction is reversible, the application of a potential between the electrodes (cathode and anode) adds energy to the circuit, and allows the process to repeat, where in single-use (primary) batteries it cannot (*1*).

The design concept of a nuclear battery involving direct energy conversion would consist of alternating layers of a direct energy converter (DEC), most likely silicon-carbide (SiC) diodes, and a nuclear isotope, which would serve as the energy source. This can be seen in figure 1. The theory is similar to the first voltaic cell (*2*), where the more layers are stacked together; the higher the voltage can be expected.
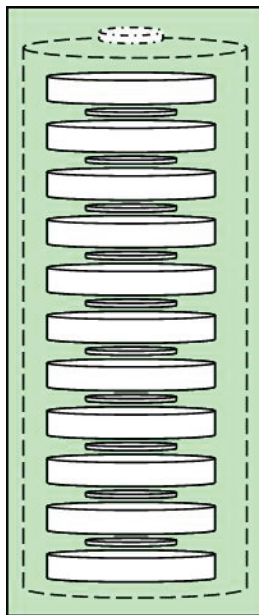
Figure 1. Alternating layers of isotope energy source and SiC direct-energy-converter (DEC) in a AA-battery package.

MCNPX is a general-purpose Monte Carlo N-Particle code which supports 34 particle types, contains cross section libraries, and includes the ability to use physics models for energies where tabular data are not available (*3*). MCNPX treats an arbitrary three-dimensional configuration of materials in geometric cells. Pointwise continuous-energy cross section data for scattering in materials are used. Source and tally structures with extensive statistical analysis of convergence are integral to the output parameters. Rapid convergence is enabled by a wide variety of variance reduction methods. MCNPX includes libraries for neutrons, photons, electrons, protons and photonuclear interactions. "Mesh" and "radiography" tallies are included for 2 and 3-dimensional imaging purposes. An auxiliary program, GRIDCONV, converts the mesh and radiography tally as well as standard mtal-file results for viewing by independent graphics packages.

Monte Carlo algorithms calculate a statistically valid, most likely results of a series of events, based on the known probability-density-functions of the events (*4*). These algorithms are useful if you want to ask what will be the outcome of a complex series of events. This is particularly useful in complex series of molecular motions or a sequential series of events as in photon scattering through a material. Each event has its own probability of occurrence. It can take days to run, in order to test as many events and outcomes as possible and keep the statistics low.

These algorithms use random number generators to create the space for variety in the complexity and follow with variance reduction techniques to develop statistics that give guidance as to the value or weight of the answer. Credit for inventing the algorithm often goes to Stanislaw Ulam.

Unlike Monte Carlo simulation, deterministic transport methods require that the phase-space be discretized in space, energy and angle, introducing a substantial degree of complication in terms of algorithm development, problem set-up, and requires a significant amount of computer memory in order to store the spatial, angular and energy dependent flux information. The large demand placed on computer memory by deterministic transport codes has limited their use. However, with the rapid increase in computer processing power, memory capacity, and reduction in unit cost, three-dimensional, time-independent, deterministic transport codes are now becoming available as production codes (*5*).

Forty-eight different isotopes have been identified in the literature, for application to nuclear batteries, as shown in table 1 (*6 through 9*). From this list, the isotope that best falls within the ideal beta-energy range can be picked for efficiency optimization. Simulations testing the ideal DEC are also described, using four different DEC's for comparison, as shown in table 2.

Table 1. Table of isotopes that have been considered in the literature for application to nuclear batteries.

A		Z	N	Halflife (yr)	alpha	beta pk (kev)	IT,ec
3	H(T)	1	2	12.32		18.6	
14	C(PMMA)	6	8	5710		156	
32	P	15	17	0.04		1711	
35	S	16	19	0.239		167.4	
42	Ar	18	24	32.9		600	
44	Ti	22	22	49.3		266	
55	Fe	26	29	2.73		232	
60	Co	27	33	5.24		310	1170,1330
63	Ni	28	35	101		66.9	
85	Kr	36	49	10.755		687	
90	Sr	38	52	28.77		546	
106	Ru	44	62	1.0234		39	
109	Cd	48	61	1.2674		184	
113	Cd	48	65	14.1		58	
121	Sn m	50	71	55		6	
137	Cs	55	82	30		190	
145	Pm	61	84	17.7		161	
147	Pm	61	86	2.624		225	
108	Ag m	47	61	418.25			109.5,2027
108	Ag g	47	61	4.50913E-06		1649	
148	Gd	64	84	75	3271		
151	Sm	62	89	90		76	
155	Eu	63	92	4.67		253	
157	Tb	65	92	99		63	

Table 1. Table of isotopes that have been considered in the literature for application to nuclear batteries (cont'd).

171	Tm	69	102	1.92		96	
178	Hf m	72	106	31			2445
179	Ta	73	106	1.79		110	
198	Au	79	119	8.56164E-08		980	
193	Pt	78	115	50		56	
204	Tl	81	123	3.78		763	
210	Pb	82	128	22.29		63	
208	Po	84	124	2.8979	5216		
210	Po	84	126	0.379	5304		
228	Ra	88	140	5.75		46	
227	Ac	89	138	21.773		44	
228	Th	90	138	1.9131	5520		
232	U	92	140	68.9	5414		
235	Np	93	142	1.085	123		
236	Pu	94	142	2.857	5867		
238	Pu	94	144	87.74	5593		
241	Pu	94	147	14.35		21	
241	Am	95	146	432	5500		
242	Am	95	147	0.001826484		645	
242	Am m	95	147	141			49
243	Am	95	148	7370	5280		
244	Cm	96	148	18.1	5902		
248	Bk	97	151	9	5793		
250	Cf	98	152	13.07	6128		

Table 2. Table of DEC that have been considered in this parametric study.

Material	Density [g/cc]
Silicon (Si)	2.3
Silicon Carbide (SiC)	3.2
cvd diamond (cvd)	3.52
Gallium Nitride GaN	6.15

This report describes a series of numerical simulations using MCNPX, with the goal of matching the characteristics of isotope energy deposition and range with DEC material properties. Fabrication of devices can be optimized to place p-n junctions in regions of maximum energy deposited by the betas from the various isotopes. This capability will increase the efficiency of the DEC electrical output for given choice of isotope.

2. Numerical Simulations

The two different parametric studies were performed in this study: 1) varying the beta-endpoint-energy while keeping the material parameters constant, and 2) varying the material while keeping the beta-endpoint-energy constant. A table of the materials and beta-endpoint-energies matrix is shown in table 3.

Table 3. Table of runs done through MXCNPX.

		DEC			
		Silicon (Si)	Silicon Carbide (SiC)	cvd diamond (cvd)	Gallium Nitride (GaN)
Beta Endpoint Energy [keV]	20		X		
	50	X	X	X	X
	100		X		
	200		X		
	500		X		

The first half of the project, which varied endpoint energy while keeping the DEC material constant, was performed by selecting various endpoint energies on a logarithmic scale in a range suitable to electron energies found in beta emitters of table 1. A discussion of the spectrum that was input for each endpoint energy, can be found in section 2.1 of this paper. The endpoint energy parameter was varied for SiC as DEC material. It is currently the most likely candidate for a nuclear battery as it is available and known to be radiation resistant. Silicon Carbide has a relatively large band gap (3.3 eV), which usually results in low leakage currents.

The second half of the parametric study varied the DEC material properties at a constant beta-endpoint energy of 50 keV. This particular energy was chosen for the material study because of its closeness in value to the endpoint energy of nickel-63 (Ni63), whose endpoint energy is 66 keV. Only an approximation is sought. Nickel-63 is an easily available isotope that may have future practical applications in experimentation.

2.1   Monte-Carlo Neutron Particle eXtended (MCNPX)

The first stage in the process of running simulations using MCNPX is to create an input deck, which consists of surfaces (shapes within the defined geometry) and cells (combinations of the surfaces). Each cell is assigned a density, as well as a material element specification, which is stated in the input deck by atomic percentage of molecular makeup, atomic number and mass. The user also specifies which particles are to be tracked through each cell.

The geometry design of the following simulations used 30 DEC layers measuring 5 mm $\times$ 5 mm $\times$ 5 μm, for a total of 150 μm of DEC stacked together. In select cases, when higher energy spectrums were used, additional simulations were run using 5 mm $\times$ 5 mm $\times$ 50 μm layers for a total coverage of 1500 μm in order to ascertain the stopping power of the DEC. In all cases, only

beta particles were tracked throughout the model, and the source was located 1mm away. The geometry of the general input deck is shown below in figure 2 for clarification.
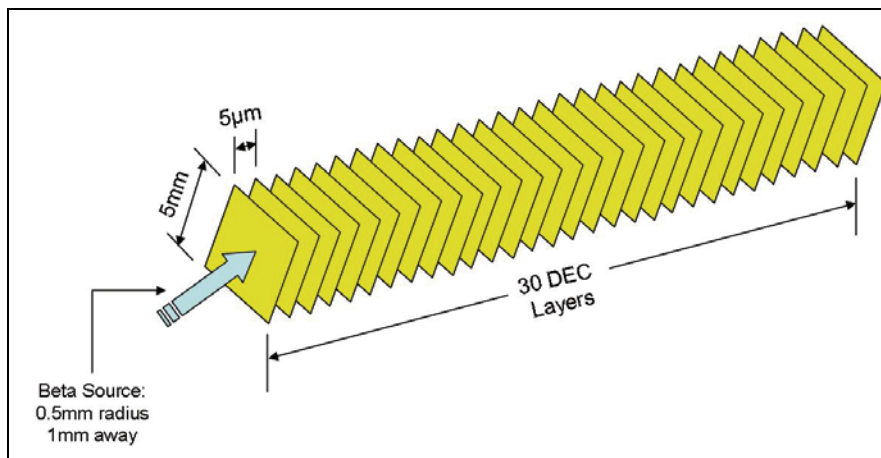
Figure 2. Model of the geometry created within MCNPX.

The beta spectrum of the simulation coded into the MCNPX input deck is based off on a Strontium-90 (Sr90) beta-emitter (*9*), and normalized for the appropriate endpoint energy. The spectrum emitted from Sr-90, like many other beta-emitters, has been shown to peak at approximately one third of the endpoint energy (*10*), although the number used for the peak in the calculations is actually less than one third. An example of one such spectrum is shown in figure 3.
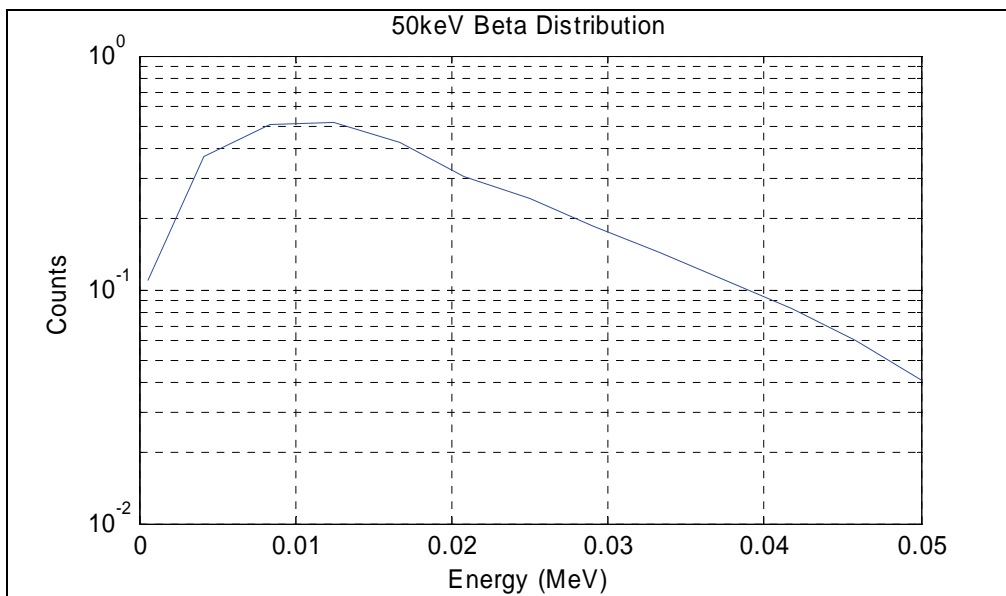
Figure 3. Beta spectra from MCNPX input deck all have the same shape, with the peak occurring just before one third of the endpoint energy. The area under the curve is normalized to one, and each varies only in endpoint. Shown above is the 50 keV beta spectrum used.

6

Although the Sr90 beta spectrum was the basis for the calculations performed, its spectrum is similar to many other isotopes' beta spectrum that may be used for future experimentation. Nickel-63 is known to vary from Sr90's spectrum, where instead of peaking at approximately one third of the endpoint energy, the peak of Ni63's output spectrum occurs at the extreme low end of the spectrum (*10*). It has been shown that the low energy electrons (<100 keV) deposit more of their energy into the materials than the high energy (>100 keV) electrons. The lower energy electrons are more likely to be captured by the depletion region of DECs and converted into usable power in a circuit (*11*). This study's purpose is to give an approximation of what to expect in future experimentation. MCNPX is a nuclear scattering code showing conversion of electron and photon incident energy into scattered electrons and photons. In parallel, a drift-diffusion circuit model is in the process of being verified. This parametric study calculates the efficiency of energy absorbed by the DEC material, as a function of isotope input-energy-spectrum and material.

The output of MCNPX includes tallies and mesh calculations of particle flux, energy deposition and dose; along with mesh calculations of similar metrics. The tallies are dumped into an output text file along with a set of default data and tables. The mesh data is dumped into a separate mesh file, and needs to be converted into text format through another program, called gridconv, which comes with the MCNPX package.

The text file outputted by MCNPX contains useful tables, including one which shows the tracks entering and the population in each layer for each particle tracked in the simulation. The tracks entering a layer show all electrons entering a layer including the source particles; if an electron exited a layer and reentered, it would be counted again. The population of a layer shows the number of tracks entering a layer plus source particles and does not include reentrant electrons, and thus population vs. tracks entering indicates backscatter. Population was useful in determining whether or not electrons were created, and up to what layer the electrons penetrated the DEC.

Delta rays, or fast moving electrons which pass through a material, cause secondary ionization. Delta rays knock orbiting electrons out of their corresponding atom, and MCNPX measures the number of free electrons generated, distinguishing them as knock-on electrons. The number of knock-on electrons produced was useful in distinguishing between properties of the endpoint energy and DEC.

There are two tallies of MCNPX which were found useful. Tally 4 measures the flux averaged over the layer and is measured in particles/cm^2, where tally 6 measures the energy deposition averaged over a cell in units of MeV/gram. Tally 4 and tally 6 are both scaled to the number of source or input particles, and have energy bins defined by the user in the input deck. The energy bins make it possible to plot the tallies as a function of energy bin and layer.

Where a tally measurement occurs in the region of a layer, a mesh measurement occurs over a grid in any defined area of the geometry of input deck. In other words, the mesh is independent of the geometry and can cover several different materials and layers if needed. There are several types of mesh measurements that are possible to make, but the one found most useful was a track averaged mesh tally, or a type 1 mesh, called pedep, which measures energy deposition in MeV/cm^3 and is also normalized by source particles. Although pedep is close to the tally 6 measurement, pedep normalizes to mesh unit volume as opposed to weight. This is because of pedep's ability to cross into regions of different densities. In both the cases of tally 6 and pedep, the amount of electron energy deposition can be used to calculate the approximate number of electrons deposited.

3. Results

There was a noticeable trend in the data gathered from the simulation of the endpoint energy study. The penetration depth of the electrons increased as expected with an increase of endpoint energy, from 5 μm in the case of 20 keV to 750 μm in the case of 500 keV. The resolution of the simulation had to be adjusted for the 200 keV and 500 keV runs from 5 μm to 50 μm in order to see the penetration depth. However, all other analysis was done for using the 5 μm layer size.

Numerical values such as population, electron activity (population – total flux), and total energy deposited were the largest in layer 1 for the 100 keV run. Population and electron activity were largest in layer 2 (5 to 10 μm) with the 500 keV endpoint, while most of the energy was deposited in layer 2 for 200 keV. The change between maximum energy deposition from 100 keV endpoint in the first layer to 200 keV endpoint in the second layer is most likely due to the stopping power of SiC, the DEC used in the simulation, to stop the lower energy betas better than the higher energy betas. However, because the low energy electrons are absorbed faster than higher energy electrons, the 20 keV run was found to have the greatest efficiency in the first layer (defined as total energy deposited in the layer/peak energy of the input electrons), and the 100 keV run was found to have the highest efficiency in the second layer. The data from the endpoint energy study is summarized in table 4(a).

Table 4. The above tables, describing the Endpoint Energy Parametric Study (a) and the Density Parametric Study (b), show a summary of results from the simulations. The results from table 4(a) show that 100keV has the highest efficiency down to the second layer of SiC. Table 4(b) shows that Si has the highest efficiency of the different materials at 50keV.

Endpoint Energy Parametric Study

Resolution	5 µm				
Number of Histories	1.E+07				
Denisty of Material (g/cc)	3.2				
Material	SiC				
Area of One Layer (cm^2) (5µm thick layers)	0.25				
Mass Per Layer (g)	0.0004				
End Point Energy	**20 keV**	**50 keV**	**100 keV**	**200 keV**	**500 keV**
Peak Energy	4.98 keV	12.5 keV	24.9 keV	49.8 keV	125 keV
Average Energy	8.2 keV	20.4 keV	40.8 keV	81.5 keV	203.8 keV
Penetration Depth	5 µm	25 µm	65 µm	200 µm	750 µm
Population in Layer 1	13467551	20522916	24131201	22824814	18019896
Flux of Layer 1 (1/cm^2/history)	0.562606	2.4465	4.4667	5.5590	5.4766
Total Flux of Layer 1	1406515	6116275	11166825	13897500	13691550
Electron Activity in Layer 1	12061036	14406641	12964376	8927314	4328346
Tracks Entering Layer 1	10000000	10215748	10922921	11534130	11593980
Total Energy Deposited in Layer 1 (MeV/g)	13.8765	31.9846	40.1016	35.7448	22.5743
Total Energy Deposited in Layer 1 (keV)	5.5506	12.79384	16.04064	14.29792	9.02972
Efficiency	68%	63%	39%	18%	4%
Population in Layer 2	0	1772867	8895986	15191634	16175905
Flux of Layer 2 (1/cm^2/history)	0	0.276493	2.19591	4.5713	5.7305
Total Flux of Layer 2	0	691233	5489775	11428175	14326225
Electron Activity in Layer 2	0	1081635	3406211	3763459	1849680
Tracks Entering Layer 2	0	762513	3920085	7763246	10425602
Total Energy Deposited in Layer 2 (MeV/g)	0	3.0785	15.411	24.5396	21.2363
Total Energy Deposited in Layer 2 (keV)	0	1.23E+00	6.16E+00	9.82E+00	8.49E+00
Efficiency	0%	6%	15%	12%	4%

(a) Endpoint Energy Study

The data gathered from the simulation of the density of the DEC had less obvious results than the endpoint energy study. However, the penetration depth was shown to increase with a decreasing density. Silicon stood out as having the highest efficiency; it both absorbed the most energy and allowed the most electrons to pass. The data from the density study is summarized in table 4(b).

Table 4. The above tables, describing the Endpoint Energy Parametric Study (a) and the Density Parametric Study (b), show a summary of results from the simulations. The results from table 4(a) show that 100keV has the highest efficiency down to the second layer of SiC. Table 4(b) shows that Si has the highest efficiency of the different materials at 50keV (cont'd).

Density Parametric Study

Resolution	5 µm			
Number of Histories	1.E+07			
Endpoint Energy	50 keV			
Peak Energy	12.5 keV			
Average Energy	20.4 keV			
Area of One Layer (cm^2) (5µm thick layers)	0.25			
Material	**Si**	**SiC**	**cvd diamond**	**GaN**
Denisty of Material (g/cc)	2.3	3.2	3.52	6.15
Mass Per Layer (g)	2.9E-04	4.0E-04	4.4E-04	7.7E-04
Penetration Depth	30 µm	25 µm	20 µm	15 µm
Population in Layer 1	19748290	20522916	19117877	22469743
Flux of Layer 1 (1/cm^2/history)	3.11999	2.4465	1.9275	1.6944
Total Flux of Layer 1	7799975	6116275	4818750	4235900
Electron Activity in Layer 1	11948315	14406641	14299127	18233843
Tracks Entering Layer 1	10404071	10215748	10119837	10032005
Total Energy Deposited in Layer 1 (MeV/g)	40.3031	31.9846	30.9761	15.8747
Total Energy Deposited in Layer 1 (keV)	16.12124	12.79384	12.39044	6.34988
Efficiency	79.03%	62.71%	60.74%	31.13%
Population in Layer 2	2830682	1772867	1723081	160593
Flux of Layer 2 (1/cm^2/history)	0.609659	0.276493	0.221778	0.0138
Total Flux of Layer 2	1524148	691233	554445	34479
Electron Activity in Layer 2	1306535	1081635	1168636	126114
Tracks Entering Layer 2	1352110	762513	751796	72939
Total Energy Deposited in Layer 2 (MeV/g)	6.41721	3.0785	3.13765	0.1240
Total Energy Deposited in Layer 2 (keV)	2.566884	1.23E+00	1.26E+00	4.96E-02
Efficiency	12.58%	6.04%	6.15%	0.24%

(b) Material Density Study

The percent energy deposition curves shown in figure 4 were created by the matlab routine, plot_edep_pk_layers.m, which is shown in appendix C. The percent energy deposition can normalizes the total energy deposition per layer with the average energy.

For the endpoint energy study, the endpoint energies of 20 keV and 50 keV stand out as leaving the most energy behind, with the 100 keV not far behind. Although the 20 keV endpoint energy data is found to deposit a significant amount of energy in the first layer, it is the only layer that the electrons penetrate. This is shown in figure 4(a).

For the material density study, the generated curves from the exponential fits are very similar, as seen in figure 4(b). Only the GaN curve deviates from the other curves with any significance, and this is most likely because its density has the greatest difference from any of the other densities. However, this deviation is small, and the similarities show that the greatest contribution of energy deposition is due largely in part to the endpoint energy of the input beta spectrum.

(a) Endpoint Energy Study   (b) Material Density Study
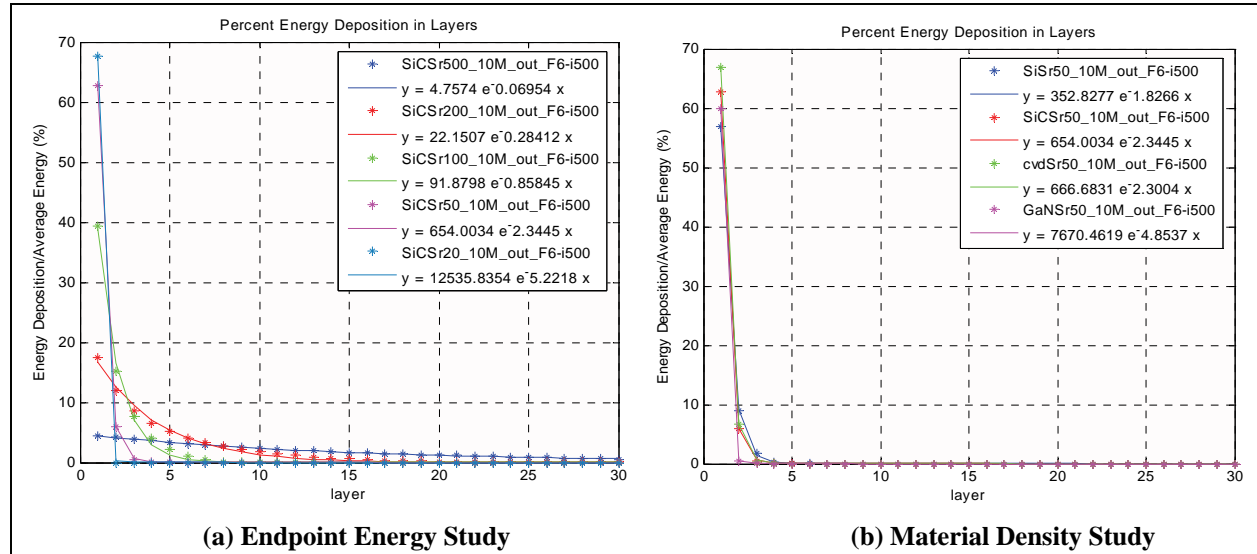
Figure 4. (a) Endpoint energies of 20 keV and 50 keV stand out as leaving the most energy behind in SiC with the 100 keV endpoint energy not far behind. (b) The similarities in the curves regardless of density show that the greatest contribution of energy deposition is due to endpoint energy. GaN is shown to vary the most from the other curves, because its density varies the most from the other densities.

More knock-on electrons are generated with higher endpoint energies on a nearly linear scale, as shown in figure 5(a). More than 10 times the number of knock-ons are created by 500 keV endpoint betas than 20 keV endpoint betas.

GaN generated the largest number of knock-on electrons of the four materials varied in the simulations. Figure 5b shows that GaN generated 7% more knock-on electrons (for 50 keV incident betas) than in SiC. While knock-on electron energies are always less energetic than the incoming electrons, the lower the energy of the knock-on, the higher probability that the energy deposited can be collected and converted to electrical current. The comparison-of-material study does not provide such clear cut results, however, as the endpoint energy study, and the number of knock-on electrons generated varies as much as 24% with the density and type of the material.

11

A valuable result is the number of energy deposited from electrons below 20 keV because the lower energy electrons are more easily collected by the junction, contributing to electrical current in the circuit. As shown in figure 6(a), more energy of this high efficient range is deposited from lower energy spectra. This is partly because the input electrons themselves are close to the ideal range, as well as the knock-on electrons produced are of a low energy. One interesting result was that although the percentage decreases, as expected with increasing endpoint energy, the maximum energy deposition within this range occurred with a 100 keV endpoint energy.

Silicon is shown to absorb the most energy at the low range in figure 6(b), which is consistent with the results listed in table 4(b). The material density results all fall within 0.5% of each other, although they do not correspond linearly, which is a common theme of the study.

(a) Endpoint Energy Study   **(b) Material Density Study**
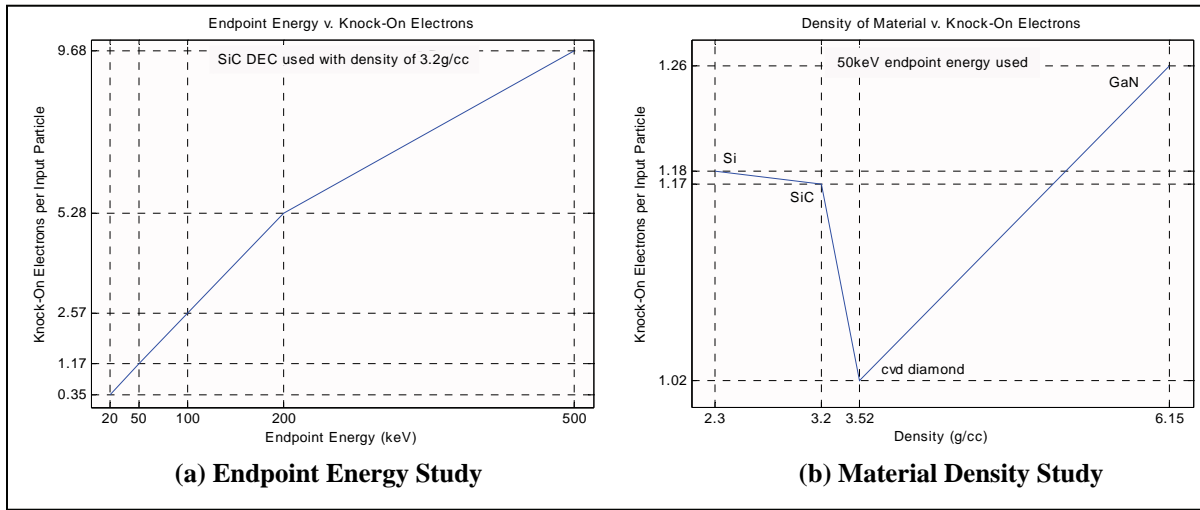
Figure 5. (a) With higher energy endpoints, more electrons are generated, but they are too high of an energy to be collected, and therefore to be useful. There is more energy lost with higher energy endpoints. (b) The density of the material seems to have no trend in its effect on the number of knock-on electrons generated.

(a) Endpoint Energy Study   **(b) Material Density Study**
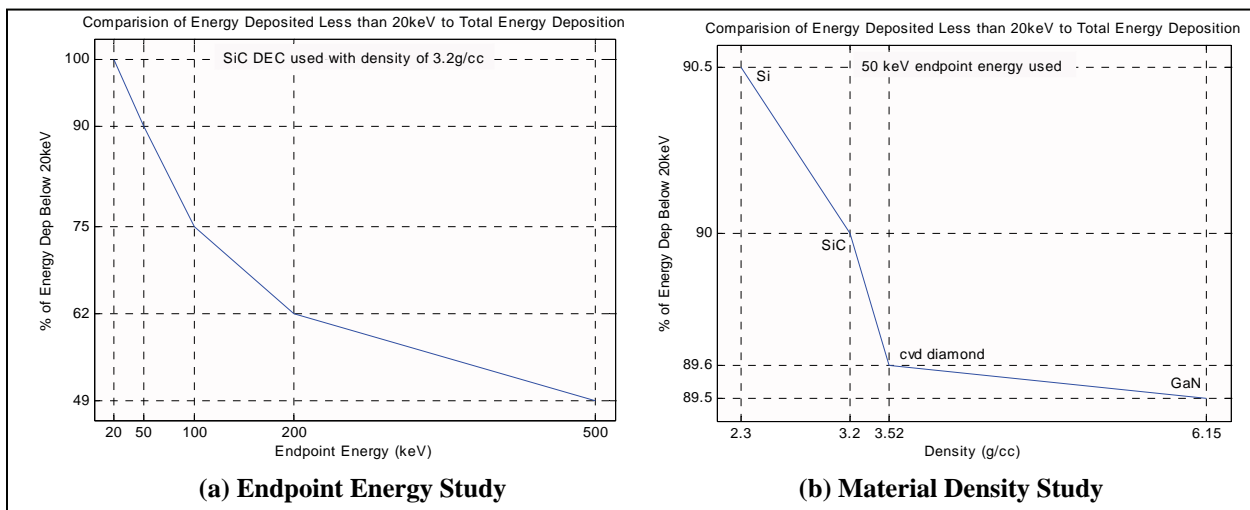
Figure 6. (a) There are more lower energy electrons deposited in the DEC as a function of the total energy deposition with lower energy endpoint energies. (b) Lower density materials absorb the energy from 20 keV electrons better than higher density materials.

4. Conclusions

Small isotope batteries (I-bat devices) would be capable of providing low levels of power for an extremely long period of time (i.e., >100 years) and would be capable of operating over a wide range of operational environments with little if any loss of performance, most notably at extremely low temperatures (i.e., < 100 K), but also in harsh biological environments. They would be ideal for distributed power applications such as MEMS devices or sensors on small (i.e., nano) sensors.

Through the use of MCNPX simulations and Matlab post-processing evaluations, we were able to discern the variation in the DEC material appears to have no definite bearing on efficiency of energy deposition. The material choice will likely play a larger role in later calculations of a drift diffusion model as well as actual experimental data. Known properties of a material will also play a role in its choice. Silicon, for example, although shown to have the highest efficiency per density, will not be a likely candidate as a DEC for a nuclear battery because of its inability to withstand degradation. At the moment, SiC seems to be the better choice because many studies have been done with SiC, and it is known to be a robust material.

The variations of energy deposition, unlike in the material study, were directly affected by beta source endpoint energy. The 20 keV endpoint energy betas provided the greatest efficiency in layer one as well as percentage of energy deposition per layer, when normalized to the average energy. However, the 20 keV betas only penetrated 5 μm into SiC, and the realistic energy output would not be high enough.

Table 5. Energy Flux compared to Energy Deposition yields an Efficiency Ratio. Though the 20 keV yields the highest ratio, the 50 keV in SiC has the highest percentage of energy deposition.

	20keV	50keV	100keV	200keV	500keV	Si	SiC	cvd diamond	GaN
Energy Flux [MeV]	0.0011	0.0113	0.0364	0.0790	0.1826	0.0142	0.0113	0.0091	0.0079
Energy Deposition [MeV]	2.78e-5	1.35e-4	2.75e-4	3.97e-4	5.44e-4	1.20e-4	1.35e-4	1.48e-4	1.29e-4
Efficiency Ratio (%)	2.492	1.194	0.755	0.502	0.298	0.844	1.194	1.625	1.632

A comparison of the energy flux and the energy deposition yields an efficiency of how much energy passes through the DEC to how much is deposited, and can be seen in table 5. The energy flux is averaged over the surface area of the layer of DEC, and because the source definition of the simulation covered approximately half of the area, the efficiency could theoretically be doubled if the source covered closer to the whole diode. The 20 keV endpoint energy generates the highest efficiency, 2.5%, but there is actually the highest percentage of energy deposited from the 50 keV endpoint energy, whose efficiency is only 1.2%. These efficiencies represent a very simplified model; it is most relevant to schottky diodes.

A closer evaluation of the results showed that endpoint energies of 50 keV or 100 keV would be ideal. The 100 keV endpoint energy betas were shown to have the highest efficiency in the second layer of SiC, or after passing through 10 μm. Both energies, however, have low penetration, making them ideal for use in Schottky diodes. The low energies with high absorption would not require additional shielding in batteries that could otherwise make them heavy and bulky.

Ni-63 falls right in the range of the idea endpoint energy, having an endpoint energy of 66 keV. According to this study, though the beta spectrum of Ni-63 varies from the beta spectrum used in the simulations, we should see high efficiencies of energy absorption. Additionally, only about 40 μm of SiC should be necessary as stopping power for Ni-63.

5.   References

1.   Buchmann, Isidor. *Batteries: in a Portable World*; Vancouver, Canada: Cadex Electronics Inc., 2001, Ch 3, 4.

2.   Letter to the Royal Society, dated March 20, 1800, Volta described the discovery of a new technique for a large voltaic pile.

3.   Briesmeister, J. F. ed. MCNP™-A General Monte Carlo N-Particle Transport Code, LA-12625-M, Version 4B (March 1997).

4.   Metropolis, Nicholas; Ulam, Stanislaw. The Monte Carlo Method. *Journal of the American Statistical Association* **1949**, *44* (247), 335–341.

5.   Kodeli, I. *Progress on Computational Tools for 3D Sensitivity and Uncertainty Analysis*. OCED-NEA Seminar on 3D Deterministic Radiation Transport Computer Programs Features, Applications and Perspectives, Paris (2-3 Dec. 1996).

6.   Sun, W.; Kherani, N. P.; Hirschman, K. D.; Gadeken, L. L.; Fauchet, P. M. A Three-Dimensional Porous Silicon p-n Diode for Betavoltaics and Photovoltaics. *Advanced Materials*, *17* (10), 1230–1233.

7.   Duggirala, R.; Li, H.; Lal, A. Active Circuits for Ultra-high Efficiency Micro-power Generators using Nickel-63 Radioisotope. Technical Digest, ISSCC, February 2006, 418–19.

8.   Lal, A.; Blanchard, J. Daintiest Dynamos. *IEEE Spectrum* **September 2004**, 36–41.

9.   Cross, W. G.; Ing, H.; Freedman, N. A short atlas of beta-ray spectra. *Phys. Med. Biol.*, *28* (11), 1251–1260.

10. Pfann, W. G.; van Roosbroeck, V. Radioactive and Photoelectric p-n junction sources. *J. Appl Phys.* **1954**, *25* (11), 1422.

11. Litz, M.; Blaine, K. *Isotope Generated Electron Density in Silicon Carbide Direct Energy Converters*; ARL-TR-3964; U.S. Army Research Laboratory: Adelphi, MD, October 2006.

Other related references: radioactive thin films for micropower

Duggirala, R.; Polcawich, A.; Zakar, E.; Dubey, M.; Lal, A. MEMS Radioisotope-powered Piezoelectric Power Generator. *IEEE MEMS 2006*, Istanbul, Turkey, January 2006, 94–97.

Chandrashekhar, C.; Thomas, C.; Li, H.; Spencer, M.; Lal, A. Demonstration of a 4H SiC Betavoltaic Cell. *2005 11th International Conference on Silicon Carbide and Related Materials*.

Radhakrishnan, S.; Guo, H.; Yoshimizu, N.; Lal, A.; Pisano, A.; Szekely, C.; McClelland, T. *On the Chip-scale Rubidium Atomic C lock*. Joint Frequency Control Symposium and Precision Timing and Time-interval Meeting, August 2005.

Radhakrishnan, S.; Lal, A. Alkali Metal-wax Micropackets for Chip-scale Atomic Clocks. IEEE Transducers 2005, June 2005, 23–26.

Lal, A.; Duggirala, R.; Li, H. Pervasive Power: A Radioisotope-powered Piezoelectric Generator. *IEEE Pervasive Computing* **Jan-Mar 2005**, *4* (1), 53–61.

Lee, C.; Guo, H.; Radhakrishnan, S.; Lal, A.; Szekely, C.; McClelland, T. A.; Pisano, A. A batch fabricated rubidium-vapor resonance cell for chip scale atomic clock. *Proceedings of the Solid State Sensor and Actuator Workshop*, Hilton Head Island, South Carolina, 2004.

Kan, Shyi-Herng; Lal, A. Self-powered IC-compatible DC-bias for Electrostatic Resonators. *Proceedings of the IEEE MEMS 2004*, Maastricht, Netherlands, 817–820.

Duggirala, R.; Li, H.; Lal, A. Radioisotope Micropower Generator for CMOS self-powered Sensor Microsystems. *Technical Digest*, PowerMEMS 2004, Kyoto, Japan 2004.

Duggirala, R.; Li, H.; Lal, A. An autonomous self-powered acoustic transmitter using radioactive thin-films. *IEEE UFFC '04*, Montreal, Canada, August 2004.

Duggirala, R.; Li, H.; Lal, A. An ultra-high efficiency piezoelectric direct charging radioisotope micropower generator. *Technical Digest, Hilton Head '04*, Hilton Head, South Carolina, USA, June, 2004.

Lal, A.; Li, H.; Guo, H. Integrated radioactive thin films for sensing systems. *Proceedings of 2004 IEEE International Symposium on Circuits and Systems (ISCAS-2004)*, Vancouver, Canada, ay 23-26, 2004, 596–599.

Guo, H.; Lal, A. Nanopower Betavoltaic Microbatteries. *Proceedings of the Transducers'03*, Boston, MA.

Li, H.; Lal, A. Radioisotape-Powered Cantilever for Vacuum Sensing with RF Transmission. *Proceedings of the Transducers'03*, Boston, MA.

Guo, H.; Lal, A. Self-Powered Photon Source. *Proceedings of the Transducers'03*, Boston, MA.

Li, Hui; Lal, A.; Blanchard, J.; Henderson, D. Self-reciprocating radioisotope-powered cantilever. *Journal of Applied Physic s* **July 2002**, *92* (2), 15, 1122–1127.

Blanchard. J. P.; Henderson, D. L.; Lal, A.; Li, H.; Guo, H.; Yao, R.; Santanam, S. Radioisotope power for MEMS devices. *Transactions of the American Nuclear Society* **2002**, *86*, 186–187.

Lal, A.; Li, H.; Blanchard, J.; Henderson, D. Direct charge radioisotope activation and power generation, (US6479920).

Li, H.; Lal, A.; Blanchard, J.; Henderson, D. Self-reciprocating Radioisotope-powered Cantilever. *Digest of Technical publications, International Conference on Solid State Sensors and Actuators*, Transducers, 2001, Munich, 744–747.

Lal, A.; Bilbao, R. M.; Leon, Y.; Guo, H.; Li, H.; Santanam, S.; Yao, R.; Blanchard. J.; Henderson, D. A Nuclear Microbattery for MEMS Devices. *Proc. 9th International Conference on Nuclear Engineering*, Nice, France, April 2001.

INTENTIONALLY LEFT BLANK

Appendix A. Additional Figures

Parametric study of a) beta-end-point energy and b) material type have been described in the body of this report. The figures graphically shown in this appendix pictorially show more of the details described in the paper. Figures A-1 through A-5 show the a) flux and b) energy deposition in the 30 SiC layers (if electrons got that far), as a function of beta endpoint energy [20, 50, 100, 200, and 500 keV]. Figures A-6 through A-8 show the variation of flux and energy deposition as a function of the material [Si, cvd diamond, GaN]. Figure A-9 shows the number density of electrons generated, in the SiC layers. The contour plots differ by the input source of the five beta endpoint energies. Figure A-10 shows penetration depth of 50 keV betas as a function of beta endpoint energy and material type. Figure A-11 shows the energy flux efficiency ratios as a function of beta endpoint energy and material type.

(a)   (b)
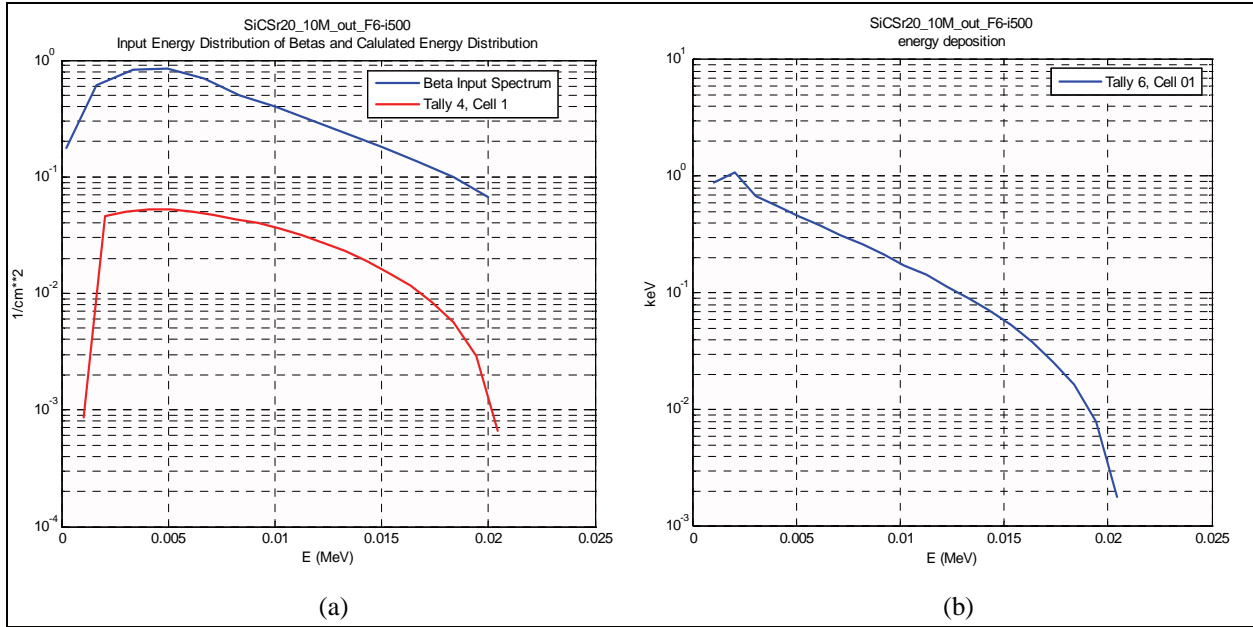
Figure A-1. (a) Shows the input beta spectrum for the 20 keV simulation as well as the resulting flux with SiC as
the DEC. (b) Shows the energy distribution for the same simulation. The data is shown for layer one
for both cases (the only penetrated layer).
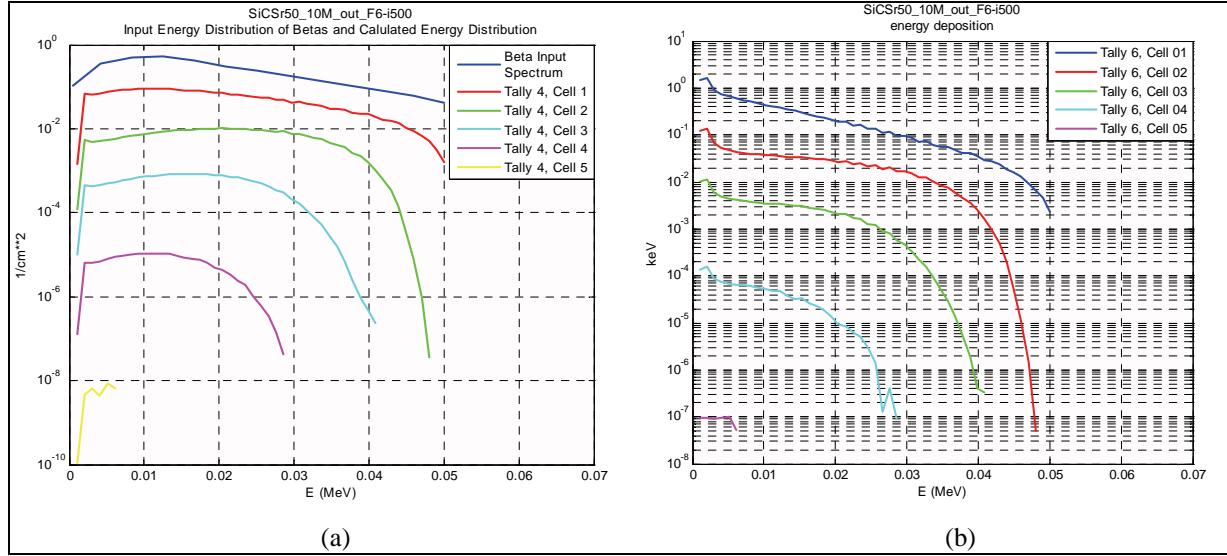
Figure A-2. (a) Shows the input beta spectrum for the 50 keV simulation as well as the resulting flux with SiC as the DEC. (b) Shows the energy distribution for the same simulation. The data is shown for layers 1-5 in both cases (the only penetrated layers).
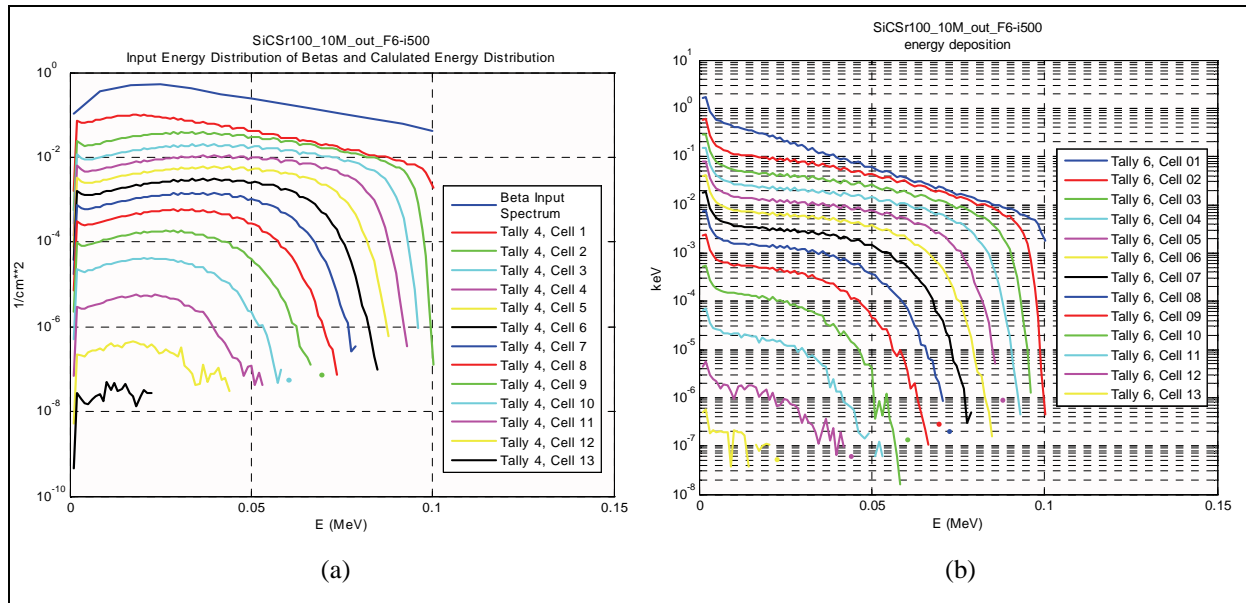
Figure A-3. (a) Shows the input beta spectrum for the 100 keV simulation as well as the resulting flux with SiC as the DEC. (b) Shows the energy distribution for the same simulation. The data is shown for layers 1-13 in both cases (the only penetrated layers).
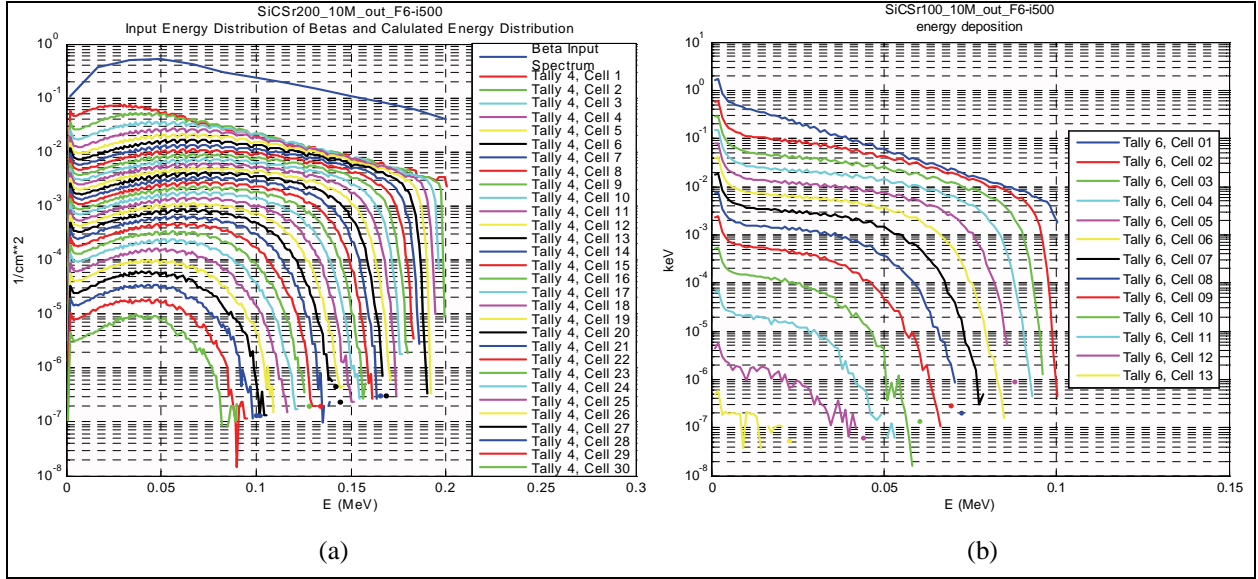
Figure A-4. (a) Shows the input beta spectrum for the 200 keV simulation as well as the resulting flux with SiC as the DEC. (b) Shows the energy distribution for the same simulation. The data is shown for all 30 layers in both cases (5 μm layer width simulation is used for best comparison).
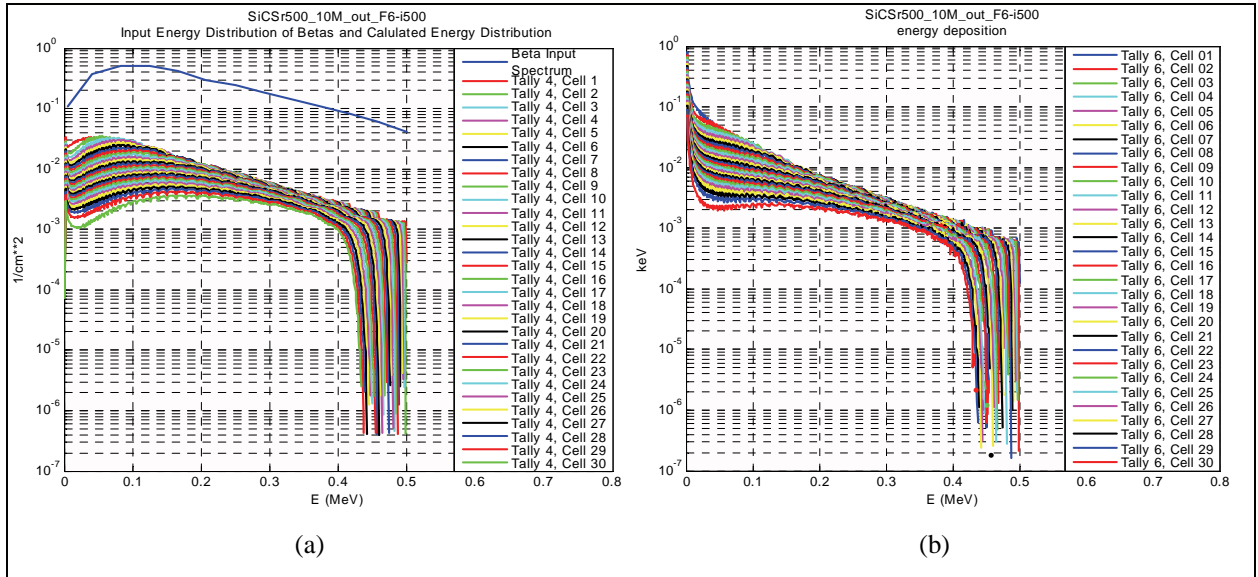

Figure A-5. (a) Shows the input beta spectrum for the 500 keV simulation as well as the resulting flux with SiC as the DEC. (b) Shows the energy distribution for the same simulation. The data is shown for all 30 layers in both cases (5 μm layer width simulation is used for best comparison).
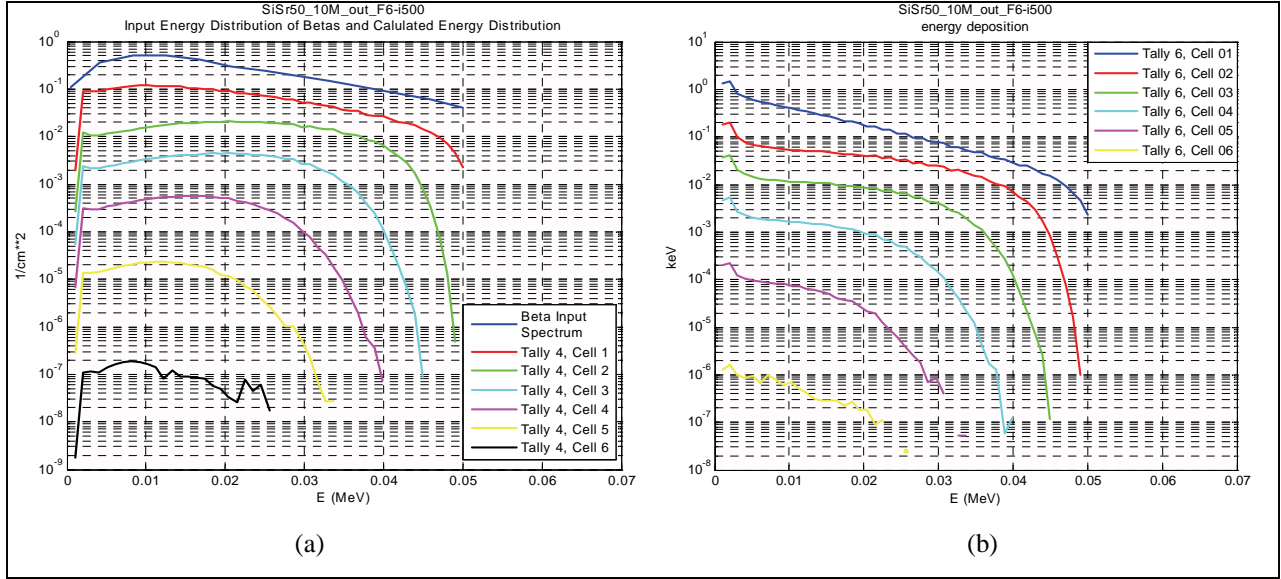
Figure A-6. (a) Shows the input beta spectrum for the 50 keV simulation with as well as the resulting flux Si as the DEC. (b) Shows the energy distribution for the same simulation. The data is shown for layers 1-6 in both cases (the only penetrated layers).
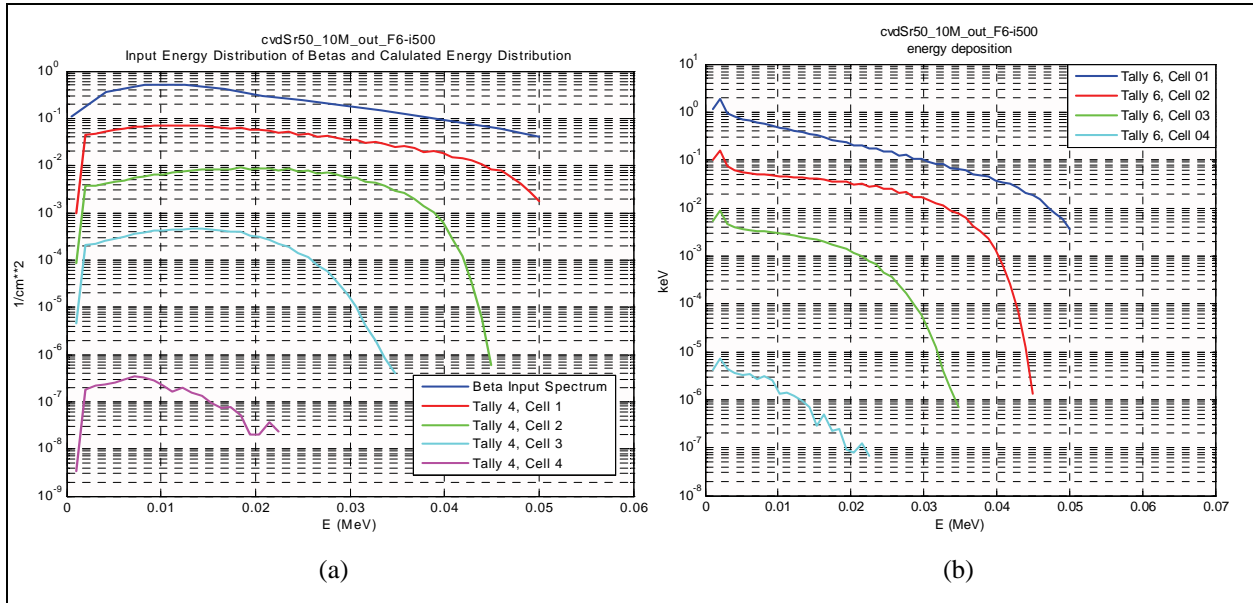

Figure A-7. (a) Shows the input beta spectrum for the 50 keV simulation as well as the resulting flux with cvd diamond as the DEC. (b) Shows the energy distribution for the same simulation. The data is shown for layers 1-6 in both cases (the only penetrated layers).
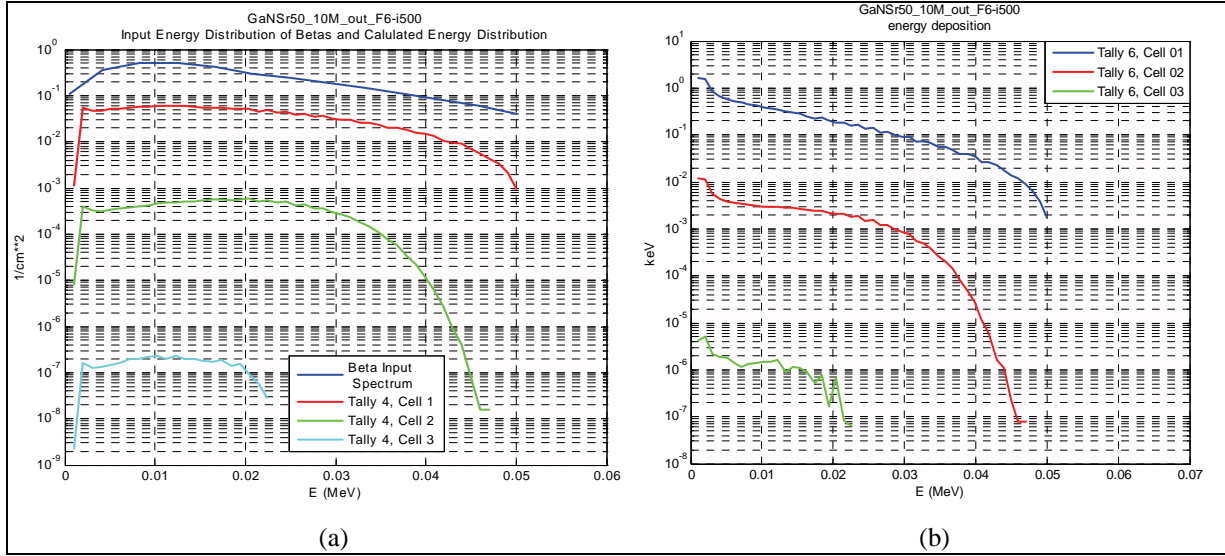
Figure A-8. (a) Shows the input beta spectrum for the 50 keV simulation as well as the resulting flux with GaN as the DEC. (b) Shows the energy distribution for the same simulation. The data is shown for layers 1-3 in both cases (the only penetrated layers).
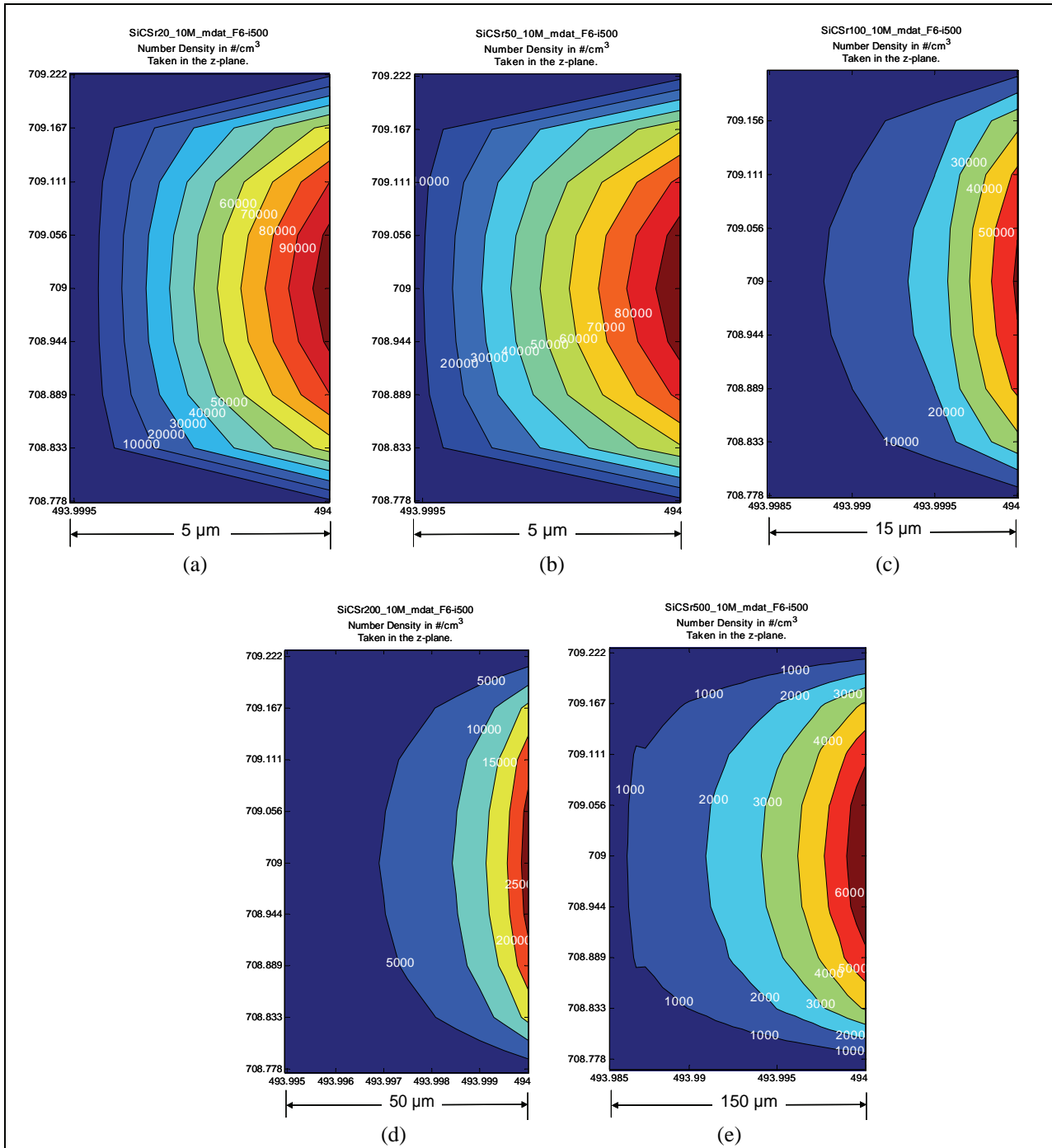
Figure A-9. Shows the mesh plots of the endpoint energy simulations. (a) 20 keV zoomed in over 5 μm (equivalent of one layer), (b) 50 keV zoomed in over 5 μm, (c) 100 keV zoomed in over 15 μm (equivalent of 3 layers), (d) 200 keV zoomed in over 50 μm (equivalent of 10 layers), and (e) 500 keV showing all 30 layers. The zoomed in region does not always denote the region in which penetration ends, only in which the mesh plot had enough variance to be interesting.
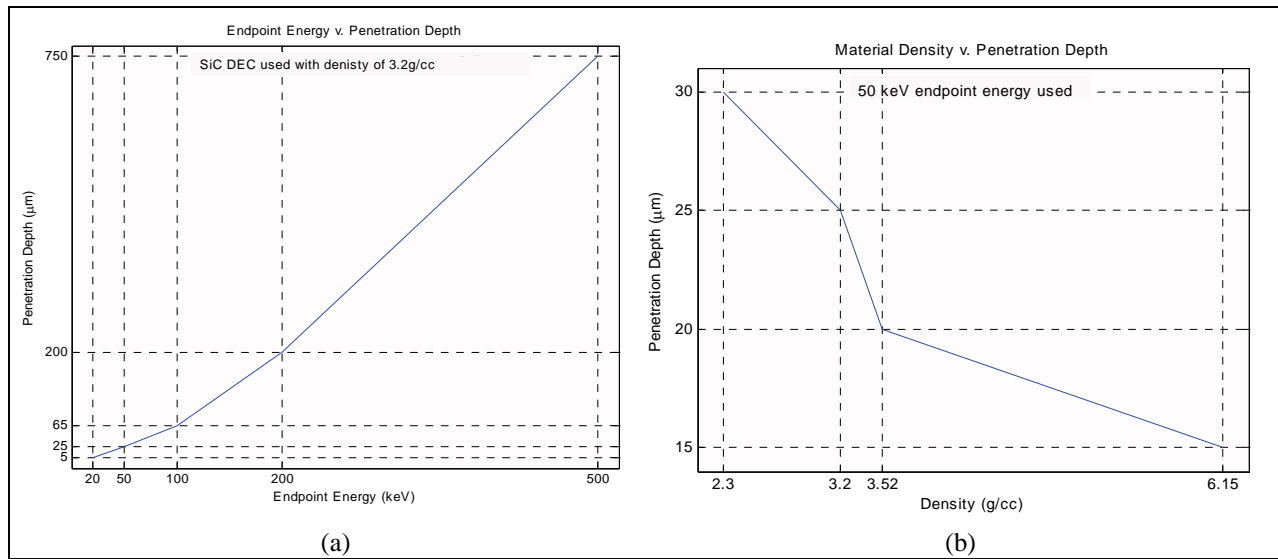
Figure A-10. These graphs show the penetration depth in the DEC as a function of (a) endpoint energy and (b) material density
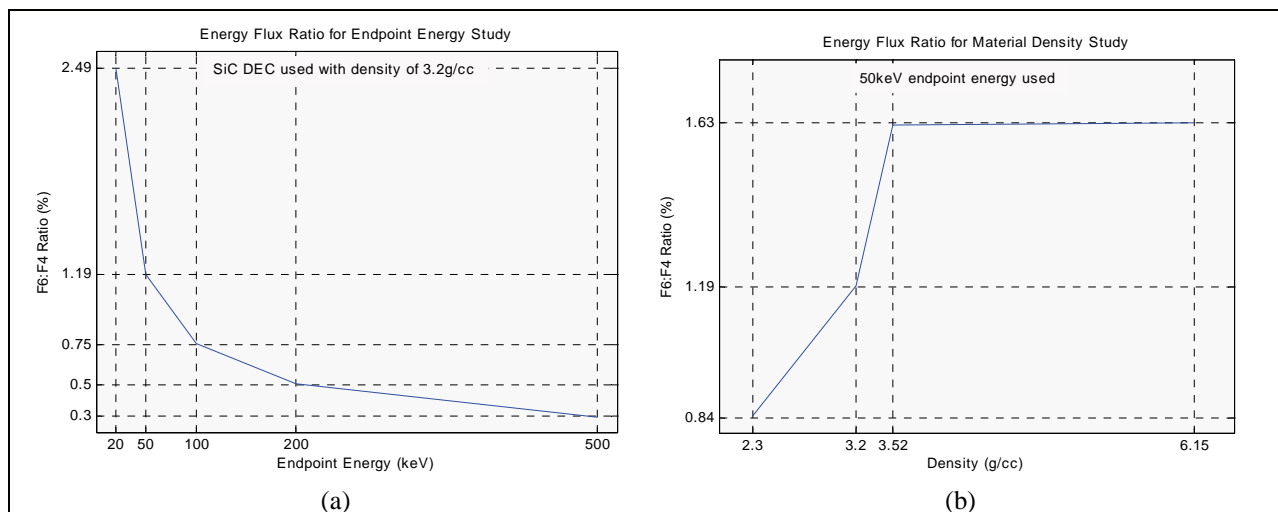
Figure A-11. Energy Flux compared to Energy Deposition yields an Efficiency Ratio. (a) Shows endpoint energy and (b) shows material density ratios.

INTENTIONALLY LEFT BLANK

Appendix B. MCNPX Input Deck

The details of the MCNPX input deck (shown below) include geometry definition statements (surfaces or volumes), materials parameters (element and density), source definition statements (beam, surface, particle type), which physics algorithms should be used during the numerical simulations, and which diagnostic results should be calculated and displayed (flux, dose, energy deposited). All of these choices have an effect on the a) complexity of the model, b) error-bars associated with the results, and c) the run-time of the simulation.

```
Model of Silicon Carbide with Sr source
c 5um layers
c (MCNPX scaled in cm)
c
c ----------------------------------------------------------
c This is a model of the way electrons would be generated
c in the SiC after being hit by electrons from a strontium
c source.
c ----------------------------------------------------------
c
c +---------------------------------------------------+
c |                                     |
c |            Cell Cards         |
c |                                     |
c +---------------------------------------------------+
c # mat  density  surface      data
c SiC
 1  3   -3.2     -1         imp:e 1  u=1   $SiC closest to
source
 2  3   -3.2     -2         imp:e 1  u=1   $2nd SiC layer
 3  3   -3.2     -3         imp:e 1  u=1   $3rd SiC layer
 4  3   -3.2     -4         imp:e 1  u=1   $4th SiC layer
 5  3   -3.2     -5         imp:e 1  u=1   $5th SiC layer
 6  3   -3.2     -6         imp:e 1  u=1   $6th SiC layer
 7  3   -3.2     -7         imp:e 1  u=1   $7th SiC layer
 8  3   -3.2     -8         imp:e 1  u=1   $8th SiC layer
 9  3   -3.2     -9         imp:e 1  u=1   $9th SiC layer
 10 3   -3.2     -10        imp:e 1  u=1   $10th SiC layer
 11 3   -3.2     -11        imp:e 1  u=1   $11th SiC layer
 12 3   -3.2     -12        imp:e 1  u=1   $12th SiC layer
 13 3   -3.2     -13        imp:e 1  u=1   $13th SiC layer
 14 3   -3.2     -14        imp:e 1  u=1   $14th SiC layer
 15 3   -3.2     -15        imp:e 1  u=1   $15th SiC layer
 16 3   -3.2     -16        imp:e 1  u=1   $16th SiC layer
 17 3   -3.2     -17        imp:e 1  u=1   $17th SiC layer
 18 3   -3.2     -18        imp:e 1  u=1   $18th SiC layer
 19 3   -3.2     -19        imp:e 1  u=1   $19th SiC layer
 20 3   -3.2     -20        imp:e 1  u=1   $20th SiC layer
 21 3   -3.2     -21        imp:e 1  u=1   $21th SiC layer
 22 3   -3.2     -22        imp:e 1  u=1   $22th SiC layer
 23 3   -3.2     -23        imp:e 1  u=1   $23th SiC layer
 24 3   -3.2     -24        imp:e 1  u=1   $24th SiC layer
```

```
 25 3   -3.2     -25        imp:e 1  u=1   $25th SiC layer
 26 3   -3.2     -26        imp:e 1  u=1   $26th SiC layer
 27 3   -3.2     -27        imp:e 1  u=1   $27th SiC layer
 28 3   -3.2     -28        imp:e 1  u=1   $28th SiC layer
 29 3   -3.2     -29        imp:e 1  u=1   $29th SiC layer
 30 3   -3.2     -30        imp:e 1  u=1   $30th SiC layer
c
c air encasing
 31 2   -0.00129  -31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 &
    20 21 22 23 24 25 26 27 28 29 30  imp:e 1  u=1
c
 32 0  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17   &
       18 19 20 21 22 23 24 25 26 27 28 29 30     &
       imp:e 1  u=1
 33 0          31       imp:e 1 u=1
c room
 34 2   -0.00129  -32      imp:e 1 fill=1
 35 0          32       imp:e 0     $outside world

c +---------------------------------------------------+
c |                                     |
c |            Surface Cards         |
c |                                     |
c +---------------------------------------------------+
c # type params
c SiC
 1  rpp  708.75 709.25  493.9995 494     117.25 117.75
c $SiC layer closest to source
 2  rpp  708.75 709.25  493.9990 493.9995 117.25 117.75
c $2nd SiC layer
 3  rpp  708.75 709.25  493.9985 493.9990 117.25 117.75
c $3rd SiC layer
 4  rpp  708.75 709.25  493.9980 493.9985 117.25 117.75
c $4th SiC layer
 5  rpp  708.75 709.25  493.9975 493.9980 117.25 117.75
c $5th SiC layer
 6  rpp  708.75 709.25  493.9970 493.9975 117.25 117.75
c $6th SiC layer
 7  rpp  708.75 709.25  493.9965 493.9970 117.25 117.75
c $7th SiC layer
 8  rpp  708.75 709.25  493.9960 493.9965 117.25 117.75
```

```
c $8th SiC layer
 9  rpp  708.75 709.25 493.9955 493.9960 117.25 117.75
c $9th SiC layer
10  rpp  708.75 709.25 493.9950 493.9955 117.25 117.75
c $10th SiC layer
11  rpp  708.75 709.25 493.9945 493.9950 117.25 117.75
c $11th SiC layer
12  rpp  708.75 709.25 493.9940 493.9945 117.25 117.75
c $12th SiC layer
13  rpp  708.75 709.25 493.9935 493.9940 117.25 117.75
c $13th SiC layer
14  rpp  708.75 709.25 493.9930 493.9935 117.25 117.75
c $14th SiC layer
15  rpp  708.75 709.25 493.9925 493.9930 117.25 117.75
c $15th SiC layer
16  rpp  708.75 709.25 493.9920 493.9925 117.25 117.75
c $16th SiC layer
17  rpp  708.75 709.25 493.9915 493.9920 117.25 117.75
c $17th SiC layer
18  rpp  708.75 709.25 493.9910 493.9915 117.25 117.75
c $18th SiC layer
19  rpp  708.75 709.25 493.9905 493.9910 117.25 117.75
c $19th SiC layer
20  rpp  708.75 709.25 493.9900 493.9905 117.25 117.75
c $20th SiC layer
21  rpp  708.75 709.25 493.9895 493.9900 117.25 117.75
c $21th SiC layer
22  rpp  708.75 709.25 493.9890 493.9895 117.25 117.75
c $22th SiC layer
23  rpp  708.75 709.25 493.9885 493.9890 117.25 117.75
c $23th SiC layer
24  rpp  708.75 709.25 493.9880 493.9885 117.25 117.75
c $24th SiC layer
25  rpp  708.75 709.25 493.9875 493.9880 117.25 117.75
c $25th SiC layer
26  rpp  708.75 709.25 493.9870 493.9875 117.25 117.75
c $26th SiC layer
27  rpp  708.75 709.25 493.9865 493.9870 117.25 117.75
c $27th SiC layer
28  rpp  708.75 709.25 493.9860 493.9865 117.25 117.75
c $28th SiC layer
29  rpp  708.75 709.25 493.9855 493.9860 117.25 117.75
c $29th SiC layer
30  rpp  708.75 709.25 493.9850 493.9855 117.25 117.75
c $30th SiC layer
c lead block to encase silicon carbide
31  rpp  708.5 709.5  493.850 494    117 118
c room dimensions
32  rpp  0 1419  0 1984  0 466


c +---------------------------------------------+
c |                                             |
c |            Material Cards           |
c |                                             |
c +---------------------------------------------+
c m# isotope    percent  &
c m1  82000    1.0     $lead 11.34 g/cc
  m2  8016     .3    &
      7014     .7    $air 0.00129 g/cc
```

```
  m3  6012       .43      &
      14028      .57    $SiC 3.2 g/cc
c m4 14028    1.0     $Si  2.3 g/cc
c m13 6012    1.0     $C   2.3 g/cc
c m5  90000     1.0     $Sr
c
c +---------------------------------------------+
c |                                             |
c |            Source Definition        |
c |                                             |
c +---------------------------------------------+
sdef       &
   dir=1     &
   erg= d2    &
   vec=0 -1 0 &
   par=e     &
   rad=d1     &
   pos=709 494.1 117.5
c
 si1 0 .1
 sp1 -21
c
c energy distribution taken from  now strontium
c (old: 280kv-8ma-12may-aft.spe)
 si2 9.16E-05 0.000549 0.00412 0.00833 0.0125 0.0167 &
0.0208 0.025 0.0291 0.0333 0.0375 0.0417 0.0459 0.05
 sp2 0     1000   3405  4692  4768 3900 2775  &
    2234  1725  1331  1007  764   556   378
c
c si3 h .707 1
c sp3 d 0 1
c
c +---------------------------------------------+
c |                                             |
c |            Data Cards               |
c |                                             |
c +---------------------------------------------+
mode e
nps 10000000    $10Mh~100min
c
c +---------------------------------------------+
c |                                             |
c |            Tallies              |
c |                                             |
c +---------------------------------------------+
c
c fc2 electron flux over a surface in particles/cm^2
c f2:e 1.4 2.4 3.4
c e2 0 98i .51
c
fc4 electron flux averaged over the cell in particles/cm^2
f4:e 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 &
    18 19 20 21 22 23 24 25 26 27 28 29 30
e4 0 498i .51
c
fc6 electron energy deposition averaged over the cell in
MeV/g
f6:e 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 &
    18 19 20 21 22 23 24 25 26 27 28 29 30
```

```
e6 0 498i .51
c
c fc8 electron energy deposition in MeV
c *f8:e 1 2 3
c e8 0 1e-5 97i .51
c
c +-----------------------------------------------------+
c |                                                     |
c |                   Mesh                   |
c |                                                     |
c +-----------------------------------------------------+
c
tmesh
c
c (a)radial meshes (b)zaxis(symmetry) (c)angles
counterclockwise
c
c  flux=#/cm2    $dose=rem/hr    $pedep=MeV/cm3
c
c flux: flux
c pedep: total energy deposition/unit vol
c dose: dose over cross planes
c
rmesh1:e  flux pedep popul dose
c layer 1 SiC
cora1   708.75   18i  709.25
corb1   493.9995     494
corc1   117.25   18i  117.75
c
rmesh21:e  flux pedep popul dose
c layer 1 SiC
cora21   708.75   18i  709.25
corb21   493.9995 18i  494
corc21   117.25       117.75
c
rmesh31:e  flux pedep popul dose
c layer 1 SiC
cora31   708.75       709.25
corb31   493.9995 18i  494
corc31   117.25   18i  117.75
c
c
rmesh41:e  flux pedep popul dose
c layer 2 SiC
cora41   708.75   18i  709.25
corb41   493.999      493.9995
corc41   117.25   18i  117.75
c
rmesh51:e  flux pedep popul dose
c layer 2 SiC
cora51   708.75   18i  709.25
corb51   493.999  18i  493.9995
corc51   117.25       117.75
c
rmesh61:e  flux pedep popul dose
c layer 2 SiC
cora61   708.75       709.25
corb61   493.999  18i  493.9995
corc61   117.25   18i  117.75

c
c
rmesh71:e  flux pedep popul dose
c layer 3 SiC
cora71   708.75   18i  709.25
corb71   493.9985     493.999
corc71   117.25   18i  117.75
c
rmesh81:e  flux pedep popul dose
c layer 3 SiC
cora81   708.75   18i  709.25
corb81   493.9985 18i  493.999
corc81   117.25       117.75
c
rmesh91:e  flux pedep popul dose
c layer 3 SiC
cora91   708.75       709.25
corb91   493.9985 18i  493.999
corc91   117.25   18i  117.75
c
c
rmesh101:e  flux pedep popul dose
c all SiC layers (30)
cora101   708.75   18i  709.25
corb101   493.985  29i  494
corc101   117.4925     117.5075
c
endmd
```

29

INTENTIONALLY LEFT BLANK

Appendix C. Matlab Routines

In order to analyze the information gained from the MCNPX simulations, routines were written in Matlab that could efficiently read in the data files. The analysis routines include:

- read_mcnpx.m
    - reads in the data from the text file
- tally_convertF6_plot.m
    - uses read_mcnpx.m to plot out the data by tally number
    - converts tally 6 from units of MeV/gram to keV
- energyDistribution.m
    - inputs and normalizes the beta/gamma spectrum in use
- plot_espec_edep_tally.m
    - calls tally_convertF6_plot.m, which calls read_mcnpx.m
    - calls energyDistribution.m
    - yields plots of tally_convertF6_plot.m as well as a plot of tally 4 data with the normalized energy spectrum
- tally_compareF6_cell1.m
    - calls read_mcnpx.m
    - plots the converted tally 6 data (in keV instead of MeV/gram) from the first layer of multiple simulations, allowing for easy comparison
- plot_edep_pk_layers.m
    - calls read_mcnpx.m
    - plots the percent energy deposition for specified files, which equals the total energy deposition for a layer divided by the energy peak of the input beta spectrum as well as a fitted exponential curve (with the equation)
- mcnpx_mesh_energy_convert.m
    - reads in the data from the mesh text file after conversion using the MCNPX gridconv
    - plots the mesh data in contour form
    - converts the energy deposition mesh from MeV/cm^3 to #/cc

The actual Matlab code follows.

read_mcnpx.m

```
function [filnam,prefix,E,T,fom,num,cell,titles,units,type,Nenergybins]=read_mcnpx
%
% format:
%        read_mcnpx
%
% input:
%        - mcnpx output file
%
% output:
%        - E : the energy bins of the various tallies stored in cell
```

```
%        arrays and matrices... for example, if there are 3 different
%        tally calls within a run, E will be a 1x3 cell array.  the size
%        of each cell is based on the number of energy bins for the
%        individual tally and the number of layers calculated in the
%        different tallies.  the benefit of cell arrays is that each cell
%        does not have to be the same size.
%        - T : the actual data taken from the tallies.  it is stored in
%        the same manner as the E array
%        - fom : the statistics corresponding to the tallies.  stored in
%        the same way as E and T.
%        - num : the tally number, corresponding to F2,4,6,8... etc.  num
%        should be the same dimensions as E, however stored as a matrix.
%        for instance, if there are 3 different tallies, num should be a
%        1x3 matrix where each entry is a number
%        - cell : cell is a cell array which holds the cell numbers of
%        the different layers corresponding to the data.  in the case of
%        3 different tallies, cell would be a 1x3 cell array, where each
%        cell would contain a 1xn matrix where n is the number of layers
%        being calculated in the tally
%        - titles : cell array which stores the titles corresponding to
%        the data
%        - units : cell array which stores the units of the data
%        - type : stores the particle type of the data in a cell array
%
% This routine has been written to be used in conjunction with other
% plotting routines.
%

%
% Pick an output file to read the data from
%
suffix='.' ;
prefix='C:\MCNPX\scratch\';
%prefix='C:\Documents and Settings\daq\Desktop\air,water,SiC\';
if nargin==0 ;filnam='outp';end
 pathfilename=prefix;
 [filnam,prefix,filterindex] = uigetfile([pathfilename,'*.*']);
 if filterindex==0,
    filnam=0; prefix=0;
    E=[]; T=[]; fom=[]; num=[]; cell=[]; titles=[]; units=[]; type=[]; Nenergybins=0;
    return;
 end;

%
% Initialize Data Matrices
%
ncell=0;    Nlines=20000;
Nenergybins=800;
% Initialize only one column, and concatenate additional columns later to
% accomodate more data
E_tally=zeros(Nenergybins,1);
T_tally=E_tally;
fom_tally=E_tally;
ruler='123456789 123456789 123456789 123456789 123456789 123456789 123456789
123456789 123456789 123456789 123456789 123456789 123456789';
```

```matlab
nlin=0;
tally_num=0;
ntot=0;
end_loop=0;

num={};
cell={};   E={};
titles={};  T={};
units={};   fom={};
type={};

%
% Read the Data
%
filnamcomp=[prefix,filnam];
fid=fopen(filnamcomp);

while end_loop==0
   line = fgetl(fid);  nlin=nlin+1;
   % Get out at end of input
   if ~isstr(line), break, end
   % Indicates that next diagnostic found
   % All tallies have this '1tally' marker
   [n,m]=size(line);
   if m==32
      stchar2=findstr('1tally',line);
      if ~isempty(stchar2)
         % Need to derive tally number from '1tally' line
         C=textscan(line,'%s %n %s %s %n');
         % We know for each case C{1}(1)='1tally'
         % However, these are still considered cells, so need to convert to matrix
         tally_num=C{2};
         % Skip two lines to get to line beginning with 'tally type'
         % **Had trouble getting program to find this string on its own**
         line = fgetl(fid);  nlin=nlin+1;
         line = fgetl(fid);  nlin=nlin+1;
         % Scan the line to read in the data (reads in the data into a cell)
         C=textscan(line,'%s');
         % We know for each case C{1}(1)='tally', C{1}(2)='type',
         % and C{1}(3)= modified tally number
         [n,m]=size(C{1});
         space=' ';
         tally_title=cell2mat(C{1}(4));
         for tmp=5:n-2, tally_title=[tally_title, space, cell2mat(C{1}(tmp))]; end
         tally_units=cell2mat(C{1}(n));
         % Read the next line to get the particle type
         line = fgetl(fid);  nlin=nlin+1;
         C=textscan(line,'%s');
         % We know for each case C{1}(1)='particle(s):'
         particle_type=cell2mat(C{1}(2));
      end
   end
   % Find Cell Number
   stchar3=findstr(' cell ',line);
```

```matlab
        if isempty(stchar3), stchar3=findstr(' surface ',line); end
        if ~isempty(stchar3)
            if stchar3==1
                C=textscan(line,'%s %n');
                cell_num(ntot+1)=C{2};
            end
        end
        % Now that we know what we are looking at, we need to look for 'energy'
        % Should only be the line after ' cell '/' surface '
        stchar=findstr('energy',line);
        if ~isempty(stchar)
            %
            % Generic Reading of Data.
            % The string 'energy' is found at the 7th position on the line
            if stchar==7
                ntot=ntot+1;
                for dlin=1:Nenergybins
                    line=fgetl(fid);  nlin=nlin+1;
                    stchar1=findstr('total',line);
                    if ntot==1
                        % First time only, find arraylegnth
                        if ~isempty(stchar1)
                            % The string 'total' is found at the 7th position on the line
                            if stchar1 == 7
                                Nenergybins=dlin-1;
                                disp(['modified Nenergybins ',num2str(Nenergybins)]);
                                E_tally=[];
                                T_tally=[];
                                fom_tally=[];
                                break
                            end
                        end
                    end
                    a=sscanf(line(1:15),'%f');
                    b=sscanf(line(16:29),'%f');
                    c=sscanf(line(30:35),'%f');
                    temp1(dlin,1)=a;
                    temp2(dlin,1)=b;
                    temp3(dlin,1)=c;
                end
                E_tally=cat(2,E_tally,temp1);
                T_tally=cat(2,T_tally,temp2);
                fom_tally=cat(2,fom_tally,temp3);
                disp( ['....data from ',num2str(ntot),'th tally, Cell number ',num2str(cell_num)]);
                disp( ['taken from tally ',num2str(tally_num),', ',num2str(tally_title),' in
',num2str(tally_units)]);
                disp( ['read.....  nlin=',num2str(nlin)]);
            end
        end
        % All tallies have '1analysis' to denote the end of the tally
        stchar4=findstr('1analysis',line);
        % restart counting for the next tally
        if ~isempty(stchar4)
            % is under the assumption that all tallys are of the same legnth
            ntot=0;
            E=[E,E_tally];       E_tally=[];
```

```
            T=[T,T_tally];        T_tally=[];
            fom=[fom,fom_tally]; fom_tally=[];
            num=[num,tally_num];
            cell=[cell,cell_num];
            titles=[titles,tally_title];
            units=[units,tally_units];
            type=[type,particle_type];
            clear cell_num;
        end
        % All tallies have 'terminated' to denote the end of the file
        stchar0=findstr('terminated',line);
        if stchar0==6, end_loop=1; end
    end
    fclose(fid);


end
```

tally_convertF6_plot.m

```
function [filnam,prefix,E,T,fom,num,cell,titles,units,type,Nenergybins]=tally_convertF6_plot
%
% This function uses the read_mcnpx.m read routine to read in the output
% from a run of mcnpx.
%
% This routine plots the tally data together in the same grouping as
% outputted by MCNPX.  For instance, if there are two separate F6 tally
% calls within the MCNPX input deck, there will be two F6 graphs.
%
% This version is based off of tally_generic_plot but converts the F6 tally
% from units of MeV/g to keV
%
% input:
%
% output:
%        - Multiple plots - the number of which depends on the number of
%          MCNPX tallies called in the input deck
%

[filnam,prefix,E,T,fom,num,cell,titles,units,type,Nenergybins]=read_mcnpx;
if filnam==0, filnam='Quit at file select'; return; end;


color_matrix = ['b';'r';'g';'c';'m';'y';'k'];
scrsz = get(0,'ScreenSize');

[n,m]=size(num);


% ------------------------
% Find out the width of the layers
% ------------------------
% Assume that the layers are 5x5mm in the other 2D
width=str2double( inputdlg('How wide were the layers in MCNPX (cm)?','Please input a value') );
volume=0.5*0.5*width;
if isempty(width),
    filnam='Quit at layer width';
    prefix=0; E=[]; T=[]; fom=[]; num=[]; cell=[]; titles=[]; units=[]; type=[]; Nenergybins=0;
```

```matlab
      return;
   end
% Find out the density of the material
density={'2.3 ','3.2 ','3.52','6.15'};
[s,v]=listdlg('PromptString','Select a density of your material:','Name','Select One',...
      'SelectionMode','single','ListSize',[160 80],'ListString',density);
if isempty(s),
   filnam='Quit at density select';
   prefix=0; E=[]; T=[]; fom=[]; num=[]; cell=[]; titles=[]; units=[]; type=[]; Nenergybins=0;
   return;
end
mass=str2double(density{s})*volume;


% ------------------------
% Convert F6 from MeV/g to keV
% ------------------------
for i=1:m
   if num{i}==6
      [q,r]=size(cell{i});
      for j=1:r %for each cell
         for k=1:q %for each energy bin
            T{i}(k,j)=T{i}(k,j)*1e3*mass;
         end
      end
   end
end


% ------------------------
% Plot Each Tally Together
% ------------------------
for i=1:m

   [q,r]=size(E{i});

   %for each different tally
      figure('Position',[1 scrsz(4)/2 scrsz(3)/2 scrsz(4)/2]);
         %for each different tally
         % Prints the plots
         semilogy(E{i}(:,1),T{i}(:,1),color_matrix(1),'LineWidth',2);
         title({[texlabel(filnam,'literal')],[titles{i}]});
         xlabel('E (MeV)');
         if num{i}==6, ylabel('keV');
         else, ylabel(units{i}); end
         text=(['Tally ', num2str(num{i}), ', Cell 0',num2str(cell{i}(1))]);
         grid on;
         hold on;
         whitebg([1,0.9841,0.9841]);
         for j=2:r
            if j==2, cc=2;
            else, cc=cc+1; end
            if cc>7, cc=1; end
            semilogy(E{i}(:,j),T{i}(:,j),color_matrix(cc),'LineWidth',2);
            if cell{i}(j)<10
               for k=1:q,
                  if k==1, legout=0; end
```

```matlab
                    if legout==0
                        if T{i}(k,j)>0
                            legout=1;
                            text=([text;'Tally ', num2str(num{i}), ', Cell 0',num2str(cell{i}(j))]);
                        end
                    end
                end
            else
                for k=1:q,
                    if k==1, legout=0; end
                    if legout==0
                        if T{i}(k,j)>0
                            legout=1;
                            text=([text;'Tally ', num2str(num{i}), ', Cell ',num2str(cell{i}(j))]);
                        end
                    end
                end
            end
        end
        legend_h=legend(text,'Location','Best');
        set(legend_h,'Color','w');
        hold off;
    end

    end
```

energyDistribution.m

```matlab
function [bin,dist]=energyDistribution(type)
%
% Asks questions to determine which preentered input spectrum to calculate
% from
%
% Energy levels taken from data entered into mcnpx input card
%
% Input
%    - Asks yes or no if there was a beta spectrum used
%    - If a beta spectrum was used, then you select the spectrum from a
%    list
%    - If beta spectrum was not used, asks if gamma spectrum was used
%        - If you say no to both, expect an error
%        - If neither was actually used, you must manually edit the
%        program to include other spectrums
%

% Find out if photons or electrons:
% Photons: m=6, Electrons, m=8
[n,m]=size(type);

if m==8
    % Find out what energy betas were used in the run
    energies={'0  ','10 ','20 ','50 ','100','200','500','546'};
    [s,v]=listdlg('PromptString','Select a Beta Endpoint Energy:','Name','Select One',...
        'SelectionMode','single','ListSize',[160 130],'ListString',energies);
    if isempty(s), bin=[]; dist=[]; return; end
```

```matlab
    if energies{s}=='10 '
        % 10keV - beta
        bin = [1.8315E-05 0.00010989 0.000824176 0.001666667 0.002490842 0.003333333
0.004166667 0.005 0.005824176 0.006666667 0.007490842 0.008333333 0.009175824 0.01];
    elseif energies{s}=='20 '
        % 20keV - beta
        bin = [3.663E-05 0.00021978 0.001648352 0.003333333 0.004981685 0.006666667
0.008333333 0.01 0.011648352 0.013333333 0.014981685 0.016666667 0.018351648 0.02];
    elseif energies{s}=='50 '
        % 50keV - beta
        bin = [9.16E-05 0.000549 0.00412 0.00833 0.0125 0.0167 0.0208 0.025 0.0291 0.0333
0.0375 0.0417 0.0459 0.05];
    elseif energies{s}=='100'
        % 100keV - beta
        bin = [0.000183 0.0011 0.00824 0.0167 0.0249 0.0333 0.0417 0.05 0.0582 0.0667 0.0749
0.0833 0.0918 0.1];
    elseif energies{s}=='200'
        % 200keV - beta
        bin = [0.000366 0.0022 0.0165 0.0333 0.0498 0.0667 0.0833 0.1 0.116 0.133 0.15 0.167
0.184 0.2];
    elseif energies{s}=='500'
        % 500keV - beta
        bin = [0.000916 0.00549 0.0412 0.0833 0.125 0.167 0.208 0.25 0.291 0.333 0.375 0.417
0.459 0.5];
    elseif energies{s}=='546'
        % 546keV - beta
        bin=[0.001 0.006 0.045 0.091 0.136 0.182 0.2275 0.273 0.318 0.364 0.409 0.455 0.501
0.546];
    else
        bin=[]; dist=[];
        return;
    end

    peak=bin(5);

    % beta dist
    dist = [0 1000 3405 4692 4768 3900 2775 2234 1725 1331 1007 764 556 378];

    % normalize the peak
    [n,m]=size(dist);
    [xi,yi,z]=int_trap(bin,dist);
    for i=1:m, dist(i)=dist(i)/z*peak; end
    return

elseif m==6
    % Find out what energy photons were used in the run
    energies={'bremsstrahlung','2.26MeV      ','no radiation  '};
    [s,v]=listdlg('PromptString','Select a Photon Spectrum:','Name','Select One',...
        'SelectionMode','single','ListSize',[160 100],'ListString',energies);
    if isempty(s), bin=[]; dist=[]; return; end

    if energies{s}=='bremsstrahlung'
        bin=[.001 .002 .005 .01 .015 .025 .05 .075 .1  .125 .15 .175 .2  .225 .25 .275 .3];
        dist=[0  100 500 1000 2000 3405 4692 4768 3900 2775 2234 1725 1331 1007 764  556
378];
```

```matlab
            peak=bin(8);

            % normalize the peak
            [n,m]=size(dist);
            [xi,yi,z]=int_trap(bin,dist);
            for i=1:m, dist(i)=dist(i)/z*peak; end
        elseif energies{s}=='2.26MeV      '
            bin=[2.26];
            dist=[1];
        else
            bin=[]; dist=[];
            return;
        end


        return
    end


    end
```

plot_espec_edep_tally.m

```matlab
function [filnam,prefix,E,T,fom,num,cell,titles,units,type,Nenergybins]=plot_espec_edep_tally
%
% This function uses the read_mcnpx.m read routine to read in the output
% from a run of mcnpx.
%
% This routine plots the tally data together in the same grouping as
% outputted by MCNPX.  For instance, if there are two separate F6 tally
% calls within the MCNPX input deck, there will be two F6 graphs.
%
% input:
%
% output:
%       - Multiple plots - the number of which depends on the number of
%         MCNPX tallies called in the input deck
%

[filnam,prefix,E,T,fom,num,cell,titles,units,type,Nenergybins]=tally_convertF6_plot;

% ------------------------
% Plot the Data
% ------------------------
color_matrix = ['b';'r';'g';'c';'m';'y';'k'];
scrsz = get(0,'ScreenSize');

[n,m]=size(num);

% ------------------------
% Plot Energy Speectrum of Betas and Calulated Energy Distribution
% ------------------------
legend_labels={};
for i=1:m
   if mod(num{i},10)==4
      [energy_bin,energy_dist]=energyDistribution(type{i});
      if ~isempty(energy_bin)
```

```matlab
                figure('Position',[1 scrsz(4)/2 scrsz(3)/2 scrsz(4)/2]);
                [q,r]=size(E{i});
                for j=1:r
                    if j==1
                        cc=1;
                        [a,b]=size(energy_dist);
                        if b==1
                            params=strcat(color_matrix(cc),'--p');
                            semilogy(energy_bin,energy_dist,params,'LineWidth',2);
                        else
                            semilogy(energy_bin,energy_dist,color_matrix(cc),'LineStyle','--
','LineWidth',2,'MarkerSize',12);
                        end
                        xlabel('E (MeV)');
                        ylabel(units{i});
                        grid on;
                        hold on;
                        whitebg([1,0.9841,0.9841]);
                        [a,b]=size(type{i});
                        if b==8, text=('Beta Input Spectrum');
                        elseif b==6, text=('Photon Input Spectrum'); end
                        legend_labels=[legend_labels;{text}];
                        cc=cc+1;
                    else, cc=cc+1; end
                    if cc>7, cc=1; end
                    semilogy(E{i}(:,j),T{i}(:,j),color_matrix(cc),'LineWidth',2);

                    for k=1:q,
                        if k==1, legout=0; end
                        if legout==0
                            if T{i}(k,j)>0
                                legout=1;
                                text=(['Tally ',num2str(num{i}),', Cell ',num2str(cell{i}(j))]);
                                legend_labels=[legend_labels;{text}];
                            end
                        end
                    end
                end
            end
        end
    end
    if ~isempty(energy_bin)
        if b==8,
            title({[texlabel(filnam,'literal')],['Input Energy Distribution of Betas and Calulated Energy
Distribution']});
        elseif b==6,
            title({[texlabel(filnam,'literal')],['Input Energy Distribution of Photons and Calulated Energy
Distribution']});
        end
        legend_h=legend(legend_labels,'Location','Best');
        set(legend_h,'Color','w');
        hold off;
    end
    legend_labels={};
end
```

```
            end

tally_compareF6_cell1.m
      function [filnam,prefix,E,T,fom,num,cell,titles,units,type,Nenergybins]=tally_compareF6_cell1
      %
      % This function uses the read_mcnpx.m read routine to read in the output
      % from a run of mcnpx.
      %
      % This routine plots the tally data from cell 1 of the F6 outputted by
      % MCNPXplot only and asks for the next file to plot on the same graph.  It
      % will continue to do so, until you say "no" instead of "yes".
      %

      color_matrix = ['b';'r';'g';'c';'m';'y';'k'];
      scrsz = get(0,'ScreenSize');

      cont=1;
      cc=1;
      find_density=1;
      legend_labels={};

      while cont>0

         % load in the information from read_mcnpx
         [filnam,prefix,E,T,fom,num,cell,titles,units,type,Nenergybins]=read_mcnpx;

         % determine the size of the data you are dealing with
         [n,m]=size(num);
         [q,r]=size(E{1});

         if cont==1
            % -----------------------
            % Find out the width of the layers
            % -----------------------
            % Assume that the layers are 5x5mm in the other 2D
            % Assume that all runs are calculated with the same width
            width=str2double( inputdlg('How wide were the layers in MCNPX (cm)?','Please input a
      value') );
            volume=0.5*0.5*width;
         end

         if find_density==1
            % -----------------------
            % Find out the density of the material
            % -----------------------
            density={'2.3 ','3.2 ','3.52','6.15'};
            [s,v]=listdlg('PromptString','Select a density of your material:','Name','Select One',...
                  'SelectionMode','single','ListSize',[160 80],'ListString',density);
            mass=str2double(density{s})*volume;

            % Do all the files have the same density - should it ask you the density
            % more than once?
            button = questdlg('Are all of the runs done with the same material (Does the density stay the
      same)?',...
```

```matlab
            'Please select yes or no','Yes','No ','Yes');
        if button=='Yes', find_density=0; end
    end


% ------------------------
% Convert F6 from MeV/g to keV
% ------------------------
for i=1:m
    if num{i}==6
        for j=1:r %for each cell
            for k=1:q %for each energy bin
                T{i}(k,j)=T{i}(k,j)*1e3*mass;
            end
        end
    end
end


% ------------------------
% Plot Each Tally Together
% ------------------------
for i=1:m
    % Only continue for Tally F6
    if num{i}==6
        % Only start a figure the 1st time
        if cont==1, figure('Position',[1 scrsz(4)/2 scrsz(3)/2 scrsz(4)/2]); end
        % Plot
        semilogy(E{i}(:,1),T{i}(:,1),color_matrix(cc),'LineWidth',2);
        % Only do the following the first time
        if cont==1
            title({'Parametric Comparison: Spectrum of Energy Deposited'});
            xlabel('E (MeV)');
            ylabel('Energy Deposited (keV)');
            grid on;
            hold on;
            whitebg([1,0.9841,0.9841]);
        end
        text=(texlabel(filnam,'literal'));
        legend_labels=[legend_labels;{text}];
    end
end


% ------------------------
% Decide if you want to plot another file on the same graph
% ------------------------
% If so, continue, if not, finish the graph and quit
button = questdlg('Do you want to plot another file on the same graph?','Please select yes or
no','Yes','No ','Yes');
if isempty(button)
    legend_h=legend(legend_labels,'Location','Best');
    set(legend_h,'Color','w');
    hold off
    cont=0;
elseif button=='No '
    legend_h=legend(legend_labels,'Location','Best');
    set(legend_h,'Color','w');
```

```matlab
        hold off
        cont=0;
    else %button=='Yes'
        cont=cont+1;
        cc=cc+1;
        if cc>6, cc=1; end
    end

  end

  end
```

plot_edep_pk_layers.m

```matlab
function plot_edep_pk_layers
%
% This function uses the read_mcnpx.m read routine to read in the output
% from a run of mcnpx.
%
% This routine plots the Percent Energy Deposition in the Layers by
% calculating the total energy deposition and dividing by the peak energy
% of the input spectrum used.
%
% input:
%       - Beta Endpoint Energy - 10/20/50/100/200/500 keV - from here the
%         peak energy can be determined
%       - Width - or thickness of the layers being used in the model.
%         Typical widths are 5e-4cm or 5e-2cm.  Check the model if you are not
%         sure.  The width is used to calculate the volume of the chip,
%         assuming that the other sides are 5mmx5mm (0.5cm).
%       - Density - The density of the chip is needed to find the mass
%         and to convert the units of the tally into keV from MeV/gram.
%             The density of
%                 SiC=3.2g/cc
%                 Si=2.3g/cc
%                 GaN=6.15g/cc
%                 cvd diamond=3.52g/cc
%
% output:
%       - A plot with one point per layer including an exponential fit.
%         The equation is listed in the legend.
%
cont=1;
cc=1;
while cont>0

    [filnam,prefix,E,T,fom,num,cell,titles,units,type,Nenergybins]=read_mcnpx;
    if filnam==0,
        disp(['                  ';'Quit at file select']);
        return;
    end;

    color_matrix = ['b';'r';'m';'g';'c''k';];
    scrsz = get(0,'ScreenSize');
```

```matlab
[n,m]=size(num);

% ------------------------
% Plot the Energy Deposition in the Layers
% ------------------------

% First Find the Tally Totals for F6
F6_total=[];
for i=1:m,
    if num{i}==6
        [q,r]=size(T{i});
        for k=1:r
            sum=0;
            for j=1:Nenergybins
                sum=sum+T{i}(j,k);
            end
            F6_total=[F6_total,sum];
        end
    end

    % Find out what energy betas were used in the run
    energies={'10','20','50','100','200','500'};
    [s,v]=listdlg('PromptString','Select a Beta Endpoint Energy:','Name','Select One',...
        'SelectionMode','single','ListSize',[160 80],'ListString',energies);
    if isempty(s)
        disp(['                    ';'Quit in energy select']);
        return;
    end;
    if s==1, peak=2.49; %10keV - energies(s)
    elseif s==2, avg=8.2;   %peak=4.98; %20keV
    elseif s==3, avg=20.4;  %peak=12.5; %50keV
    elseif s==4, avg=40.8;  %peak=24.9; %100keV
    elseif s==5, avg=81.5;  %peak=49.8; %200keV
    elseif s==6, avg=203.8; %peak=125;  %500keV
    end

    % Find out the width of the layers
    % Assume that the layers are 5x5mm in the other 2D
    width=str2double( inputdlg('How wide were the layers in MCNPX (cm)?','Please input a
value') );
    if isempty(width)
        disp(['                    ';'Quit in width select']);
        return;
    end
    volume=0.5*0.5*width;

    % Find out the density of the material
    density={'2.3 ','3.2 ','3.52','6.15'};
    [t,w]=listdlg('PromptString','Select a density of your material:','Name','Select One',...
        'SelectionMode','single','ListSize',[160 80],'ListString',density);
    if isempty(t),
        disp(['                    ';'Quit in density select']);
        return;
    end
    mass=str2double(density{t})*volume;
```

```matlab
%Convert F6_total from MeV/g to keV
ratio=[];
for l=1:r
    edep=F6_total(l)*1e3*mass;
    %ratio(l)=edep/peak*100;
    ratio(l)=edep/avg*100;
end

% Plot and Fit the Data
if cont==1
    figure('Position',[1*scrsz(3)/5 4*scrsz(4)/9 4*scrsz(3)/9 4*scrsz(4)/9]);
end
[estimates, model] = fitcurve(cell{i},ratio);
properties=strcat(['*',color_matrix(cc)]);
plot(cell{i},ratio,properties);
if cont==1
    hold on
    grid on
    xlabel('layer');
    ylabel('Energy Deposition/Average Energy (%)');
    title('Percent Energy Deposition in Layers');
end
[sse, FittedCurve] = model(estimates);
plot(cell{i}, FittedCurve, color_matrix(cc))

data{cont}=texlabel(filnam,'literal');
interp{cont}=texlabel(strcat(['y = ',num2str(estimates(1)),' e^',num2str(-1*estimates(2)),'
x',]),'literal');

% Changes the background color to tan but later change the legend back to white
whitebg([1,0.9841,0.9841]);

% Decide if you want to plot another file on the same graph
% If so, continue, if not, finish the graph and quit
button = questdlg('Do you want to plot another file on the same graph?','Please select yes
or no','Yes','No ','Yes');
    if isempty(button)
        if cont==1
            title({[texlabel(filnam,'literal')],...
                ['Percent Energy Deposition in Layers']});
            legend_h = legend('Data',texlabel(strcat(['y = ',num2str(estimates(1)),' e^',num2str(-
1*estimates(2)),' x',]),'literal'));
            set(legend_h,'Color','w');
        else
            for i=1:cont
                if i==1, textstring={[data{i}],[interp{i}]};
                else, textstring=cat(2,textstring,{[data{i}],[interp{i}]}); end
            end
            legend_h = legend(textstring);
            set(legend_h,'Color','w');
        end
        hold off
        cont=0;
    elseif button=='No '
        if cont==1
```

45

```matlab
                title({[texlabel(filnam,'literal')],...
                    ['Percent Energy Deposition in Layers']});
                legend_h = legend('Data',texlabel(strcat(['y = ',num2str(estimates(1)),' e^',num2str(-
1*estimates(2)),' x',]),'literal'));
                set(legend_h,'Color','w');
            else
                for i=1:cont
                    if i==1, textstring={[data{i}],[interp{i}]};
                    else, textstring=cat(2,textstring,{[data{i}],[interp{i}]}); end
                end
                legend_h = legend(textstring);
                set(legend_h,'Color','w');
            end
            hold off
            cont=0;
        else %button=='Yes'
            cont=cont+1;
            cc=cc+1;
            if cc>6, cc=1; end
        end
      end
    end

  end

  end
```

mcnpx_mesh_energy_convert.m

```matlab
function mcnpx_mesh_energy_convert
%
% Read mesh diagnostic output ....  mdat file type after ascII conversion
% Plot contours of energy flux or dose, and the associated statistics
% Logic inserted for any rectangular mesh
%       - organizes data into a 3D matrix representing x-y-z
%       - looks at size of each dimension and selects biggest side of
%         rectangle for the plane, under the assumption that there is
%         only one division in the other dimension
%       - labels on graph tell you which plane you are looking at, as
%         well as the location of the corresponding value
%          - the mcnpx mesh output only has an accuracy of 3 decimal
%            places so for smaller numbers, what looks like an error
%            of a repeating number in the data lables is actually
%            what the mdat file reads
%
% Format:
%       mcnpx_contour_read_mesh
%
% Inputs:
%       - conversion_print: if the conversion_print if inputed with any number greater
%         than zero, the conv  erted graph will be plotted in number
%         denisty #/cc
%
%       - statistics_print: if statistics_print is inputed with any number greater
%         than zero, the statisticswill be outputed as well as just the data
%
```

```matlab
%         - data_print: if data_print is inputed with any number greater
%           than zero, the data will be outputed
%
% Development
%         - labels of graph should read from the text and converted into
%           numbers, dived up manually into tick marks
%         (currently read from labels in lines 199-209)
%                 - the labels of the countour plot is not always
%                   accurate
%
% flux in #/cm2/s
% dose in rem/hr

ncell=0;   Ncells=56;  Nenergybins=800;  Nlines=20000;
E=zeros(Nenergybins,Ncells); MeVpg=E; fom=E;
ruler='123456789 123456789 123456789 123456789 123456789 123456789 123456789
123456789 123456789 123456789 123456789 123456789 123456789';


%
% *** read from mdat data ***
%
nlin=0; ncell=0;
suffix='.' ;
prefix='C:\mcnpx\scratch\';
pathfilename=prefix;
[filnam,prefix,filterindex] = uigetfile([pathfilename,'*.*']);
if filterindex==0,
    disp(['                  ';'Quit at file select'])
    return;
end;
disp([prefix,filnam])
filnamcomp = [prefix,filnam];
fid = fopen(filnamcomp);


% find the peak energy
energies={'10          ','20          ','50          ',...
    '100         ','200         ','500         ','546          ',...
    'bremsstrahlung'};
[s,v]=listdlg('PromptString','Select an Endpoint Energy:','Name','Select One',...
    'SelectionMode','single','ListSize',[160 120],'ListString',energies);
if isempty(s)
    disp(['                  ';'Quit in energy select'])
    return;
end;
if energies{s}=='10          ', peak=2.49;
elseif energies{s}=='20          ', peak=4.98;
elseif energies{s}=='50          ', peak=12.5;
elseif energies{s}=='100         ', peak=24.9;
elseif energies{s}=='200         ', peak=49.8;
elseif energies{s}=='500         ', peak=125;
elseif energies{s}=='546          ', peak=136;
elseif energies{s}=='bremsstrahlung', peak=75; end
peak=peak*1e-3; %convert peak from keV to MeV
```

```matlab
conversion_print = questdlg('Do you want to plot the converted data?','Please select yes or
no','Yes','No ','Yes');
data_print = questdlg('Do you want to plot the data before conversion?','Please select yes or
no','Yes','No ','No ');
statistics_print = questdlg('Do you want to plot the statistics of the data?','Please select yes or
no','Yes','No ','No ');



%
% input line detemines how many contours
%
header = fgetl(fid);
input = fgetl(fid);
line = fgetl(fid);
ncontours = str2num(input(1:3));

i=1;
while (i<=ncontours)
    %
    % determine size of different contours
    %
    A = fscanf(fid,'%i',19);
    B = fscanf(fid,'%g',4);  %need to step through floating numbers\
    if A(2)==2, ptype{i}='photon';
    elseif A(2)==3, ptype{i}='electron';
    else, ptype{i}=''; end
    xdim{i} = A(3);
    ydim{i} = A(4);
    zdim{i} = A(5);
    totdim{i} = A(7);
    skip{i}=0;
    for j=10:19
        % check if there are more meshes written on one line to be
        % accounted for
        if A(j)>0 & j>10,
            ncontours=ncontours+1;
            ptype{i+1}=ptype{i};
            xdim{i+1}=xdim{i};
            ydim{i+1}=ydim{i};
            zdim{i+1}=zdim{i};
            totdim{i+1}=totdim{i};
            skip{i+1}=1;
            i=i+1;
        end
        if A(j)==1, type{i}='Flux in #/cm^2 per souce particle';
        elseif A(j)==3, type{i}='Population in weight times the track length';
        elseif A(j)==4, type{i}='Average Energy Deposition per Unit Volume in MeV/cm^3 per source
particle';
        elseif A(j)==5, type{i}='Dose in rem/hr per source particle';
        elseif A(j)==0, %don't do anything
        else type{i}='';
        end
    end
    i=i+1;
```

```matlab
end

for i=1:ncontours
    %
    % determine labeling
    %
    if skip{i}==0
        xlabels{i} = fscanf(fid,'%13g',xdim{i});
        ylabels{i} = fscanf(fid,'%13g',ydim{i});
        zlabels{i} = fscanf(fid,'%13g',zdim{i});
    else
        xlabels{i} = xlabels{i-1};
        ylabels{i} = ylabels{i-1};
        zlabels{i} = zlabels{i-1};
    end

    %
    % read data and statistics and store in temporary locations before
    % parsing
    %
    tmp_data = fscanf(fid,'%13g',totdim{i});
    tmp_stat = fscanf(fid,'%13g',totdim{i});

    data{i} = zeros(xdim{i}-1,ydim{i}-1,zdim{i}-1);
    stat{i} = zeros(xdim{i}-1,ydim{i}-1,zdim{i}-1);

    index = 1;
    for k=1:zdim{i}-1
        for j=1:ydim{i}-1
            for ii=1:xdim{i}-1
                %sprintf('%i %i %i %i',index,ii,j,k)
                data{i}(ii,j,k) = tmp_data(index);
                stat{i}(ii,j,k) = tmp_stat(index);
                index = index + 1;
            end
        end
    end

end
fclose(fid);

for i=1:ncontours
    [n,m]=size(type{i});
    % The string:
    % 'Average Energy Deposition per Unit Volume in MeV/cm^3 per source particle'
    %         is 73 characters long, and we are searching for it - we
    %         created type{i} ourselves based on a number from the mdat file
    if m==73
        %
        % surface plot can only handle a 2x2 matrix
        % if zdim=2 (only one z-point), then this does not matter
        % if zdim is important, then it does
        % determine which dimensions to look at
        % possibly break up into more than one contour plot
        %
```

```matlab
sdim=zdim{i}; idim=xdim{i}; jdim=ydim{i}; order='xy'; plane='z';
if ydim{i}<sdim, sdim=ydim{i}; idim=xdim{i}; jdim=zdim{i}; order='xz'; plane='y'; end
if xdim{i}<sdim, sdim=xdim{i}; idim=ydim{i}; jdim=zdim{i}; order='yz'; plane='x'; end

contour = zeros(idim-1,jdim-1,sdim-1);
statistics = zeros(idim-1,jdim-1,sdim-1);

if order=='xz'
    contour = permute(data{i},[1 3 2]);
    statistics = permute(stat{i},[1 3 2]);
    ilabel=xlabels{i}; Iaxis='X-Axis';
    jlabel=zlabels{i}; Jaxis='Z-Axis';
    %     klabel=ylabels{i};
elseif order=='yz'
    contour = permute(data{i},[2 3 1]);
    statistics = permute(stat{i},[2 3 1]);
    ilabel=zlabels{i}; Iaxis='Z-Axis';
    jlabel=ylabels{i}; Jaxis='Y-Axis';
    %       klabel=xlabels{i};
else
    contour = data{i};
    statistics = stat{i};
    ilabel=ylabels{i}; Iaxis='Y-Axis';
    jlabel=xlabels{i}; Jaxis='X-Axis';
    %       klabel=zlabels{i};
end
%
% since we only want to make a 2D plot at this point, drop all other
% layers of contour and statistics except the first in the sdim direction.
% in most cases, there is only one layer anyway, so there are no plots
% being neglected
%
% it is important to only have a 2D array for the contourf plotting
% function
%
contour = contour(:,:,1);
statistics = statistics(:,:,1);

% Divide up the labels into 10 and 5 tick marks
[ni,mi]=size(ilabel);
[nj,mj]=size(jlabel);
if ni>5, scale_factor_i = int32(ni/5);,
else scale_factor_i = 1; end
if nj>10, scale_factor_j = int32(nj/10);
else scale_factor_j = 1; end
iaxis = [ilabel(1)];
jaxis = [jlabel(1)];
for iii=1+scale_factor_i:scale_factor_i:ni, iaxis = [iaxis,ilabel(iii)]; end
for jj=1+scale_factor_j:scale_factor_j:nj, jaxis = [jaxis,jlabel(jj)]; end

%
% plot contour and statistics
% new plot for each energy distribution mesh
%
```

```matlab
        % Move button commands into actions
        if isempty(conversion_print), cp=0;
        elseif conversion_print=='No ', cp=0;
        else, cp=1; end

        if isempty(data_print), dp=0;
        elseif data_print=='No ', dp=0;
        else, dp=1; end

        if isempty(statistics_print), sp=0;
        elseif statistics_print=='No ', sp=0;
        else, sp=1; end

        % Always print something
        if (cp==0) & (dp==0) & (sp==0), cp=1; end

        scrsz = get(0,'ScreenSize');
        if cp==1
            figure('Position',[scrsz(3)/3 scrsz(2) scrsz(3)/3 2*scrsz(4)/3]);
            [C,h] = contourf( contour/peak );
clabel(C,h,'FontSize',12,'Color','w','LabelSpacing',150,'Rotation',0);
            %[h] = imagesc( contour/peak ); colorbar;
            title({[texlabel(filnam,'literal')],...
                ['Number Density in #/cm^3'],...
                ['Taken in the ',plane,'-plane.']});
            set(gca,'XTickLabel',iaxis)
            set(gca,'YTickLabel',jaxis)
            xlabel(Iaxis);
            ylabel(Jaxis);
        end
        if dp==1
            figure('Position',[1 scrsz(2) scrsz(3)/3 2*scrsz(4)/3]);
            [C,h] = contourf( contour/peak );
clabel(C,h,'FontSize',12,'Color','w','LabelSpacing',150,'Rotation',0);
            %[h] = imagesc( contour/peak ); colorbar;
            title({[texlabel(filnam,'literal')],...
                [type{i}],...
                ['Taken in the ',plane,'-plane.']});
            set(gca,'XTickLabel',iaxis)
            set(gca,'YTickLabel',jaxis)
            xlabel(Iaxis);
            ylabel(Jaxis);
        end
        if sp==1
            figure('Position',[2*scrsz(3)/3 scrsz(2) scrsz(3)/3 2*scrsz(4)/3]);
            [C,h] = contourf( contour/peak );
clabel(C,h,'FontSize',12,'Color','w','LabelSpacing',150,'Rotation',0);
            %[h] = imagesc( contour/peak ); colorbar;
            title({[texlabel(filnam,'literal')],...
                ['Statistics '],...
                ['Taken in the ',plane,'-plane.']});
            set(gca,'XTickLabel',iaxis)
            set(gca,'YTickLabel',jaxis)
            xlabel(Iaxis);
            ylabel(Jaxis);
```

```
            end
        end
    end
```

calc_energy_flux_ratio.m

```matlab
function [eflux,edep,ratio]=calc_energy_flux_ratio
%
% This function uses the read_mcnpx.m read routine to read in the output
% from a run of mcnpx.
%
% This function calculates the energy flux - goes from #/cm2 to MeV/cc
%

% ------------------------
% Read in the Data
% ------------------------
[filnam,prefix,E,T,fom,num,cell,titles,units,type,Nenergybins]=read_mcnpx;
if filnam==0,
    disp(['                    ';'Quit at file select']);
    return;
end;

% ------------------------
% Find out the width of the layers
% ------------------------
width=str2double( inputdlg('How wide were the layers in MCNPX (cm)?','Please input a value') );
if isempty(width)
    disp(['                    ';'Quit in width select']);
    return;
end
% Assume that the layers are 5x5mm in the other 2D
area=0.5*0.5;
volume=area*width;

% ------------------------
% Find out the density of the material
% ------------------------
den={'2.3 ','3.2 ','3.52','6.15'};
[t,w]=listdlg('PromptString','Select a density of your material:','Name','Select One',...
    'SelectionMode','single','ListSize',[160 80],'ListString',den);
if isempty(t),
    disp(['                    ';'Quit in density select']);
    return;
end
density=str2double(den{t});
mass=density*volume;

% ------------------------
% Convert from #/cm2 to MeV/cc and find the sums
% ------------------------
[n,m]=size(num);
F4_total=[]; F6_total=[];

for i=1:m
    [q,r]=size(T{i});
```

```matlab
        if mod(num{i},10)==4
            for k=1:r
                sum=0;
                for j=1:Nenergybins
                    %go from units of #/cm2 to MeV/cm2 to MeV/cc to MeV/g
                    %T{i}(j,k)=T{i}(j,k)*E{i}(j,k)/width/density;
                    %go from units of #/cm2 to MeV/cm2 to MeV
                    T{i}(j,k)=T{i}(j,k)*E{i}(j,k)*area;
                    %calculate the sum
                    sum=sum+T{i}(j,k);
                end
                %store the sums from all the cells in an array
                F4_total=[F4_total,sum];
            end
        elseif mod(num{i},10)==6
            for k=1:r
                sum=0;
                for j=1:Nenergybins
                    %go from MeV/g to MeV
                    T{i}(j,k)=T{i}(j,k)*E{i}(j,k)*mass;
                    %calculate the sum
                    sum=sum+T{i}(j,k);
                end
                %store the sums from all the cells in an array
                F6_total=[F6_total,sum];
            end
        end
end

% ------------------------
% For the first cell only, compare the sums, and find the percentage
% difference (assume that the cell number is the same)
% ------------------------
eflux=F4_total(1);
edep=F6_total(1);
ratio=edep/eflux*100;
disp(['F4 and F6 tallies have a ',num2str(ratio),'% ratio']);

end
```

Distribution List

ADMNSTR
DEFNS TECHL INFO CTR
ATTN DTIC-OCP (ELECTRONIC COPY)
8725 JOHN J KINGMAN RD STE 0944
FT BELVOIR VA 22060-6218

DARPA
ATTN IXO S WELBY
3701 N FAIRFAX DR
ARLINGTON VA 22203-1714

OFC OF THE SECY OF DEFNS
ATTN ODDRE (R&AT)
THE PENTAGON
WASHINGTON DC 20301-3080

US ARMY RSRCH DEV AND ENGRG CMND
ARMAMENT RSRCH DEV AND ENGRG
CNTR
ARMAMENT ENGRG AND TECHNLTY CTR
ATTN AMSRD-AAR-AEF-T J MATTS
BLDG 305
ABERDEEN PROVING GROUND MD
 21005-5001

US ARMY TRADOC
BATTLE LAB INTEGRATION & TECHL
DIRCTRT
ATTN ATCD-B
10 WHISTLER LANE
FT MONROE VA 23651-5850

PM TIMS, PROFILER (MMS-P) AN/TMQ-52
ATTN B GRIFFIES
BUILDING 563
FT MONMOUTH NJ 07703

SMC/GPA
2420 VELA WAY STE 1866
EL SEGUNDO CA 90245-4659

COMMANDING GENERAL
US ARMY AVN & MIS CMND
ATTN AMSAM-RD W C MCCORKLE
REDSTONE ARSENAL AL 35898-5000

US ARMY INFO SYS ENGRG CMND
ATTN AMSEL-IE-TD F JENIA
FT HUACHUCA AZ 85613-5300

ECOPULSE INC
7844 VERVAIN CT
SPRINGFIELD VA 22150

US GOVERNMENT PRINT OFF
DEPOSITORY RECEIVING SECTION
ATTN MAIL STOP IDAD J TATE
732 NORTH CAPITOL ST., NW
WASHINGTON DC 20402

US ARMY RSRCH LAB
ATTN AMSRD-ARL-CI-OK-TP TECHL LIB
 T LANDFRIED (2 COPIES)
BLDG 4600
ABERDEEN PROVING GROUND MD
 21005-5066

DIRECTOR
US ARMY RSRCH LAB
ATTN AMSRD-ARL-RO-EV W D BACH
PO BOX 12211
RESEARCH TRIANGLE PARK NC 27709

US ARMY RSRCH LAB
ATTN AMSRD-ARL-D J M MILLER
ATTN AMSRD-ARL-CI-OK-T
 TECHL PUB (2 COPIES)
ATTN AMSRD-ARL-CI-OK-TL
 TECHL LIB (2 COPIES)
ATTN AMSRD-ARL-SE-DE K BLAINE
ATTN AMSRD-ARL-SE-DE M LITZ
ATTN IMNE-ALC-IMS MAIL & RECORDS
MGMT
ADELPHI MD 20783-1197